Name:- Vinit Ramrao Jadhav

PRN :- 202401040163

Roll No :- CS2-26

Topic:- House Price

EDS Theory Activity 1

PROGRAM

```
import pandas as pd


# Create DataFrame (removing duplicate SqFt column)
house_price_data = {
   "ID": [1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010],
   "Address": [
      "123 Oak St, Seattle", "456 Maple Ave, Portland", "789 Pine Rd, Austin",
      "321 Elm Blvd, Denver", "654 Cedar Ln, Boston", "987 Birch Dr, Atlanta",
      "135 Spruce Way, Chicago", "864 Willow Cir, Dallas", "249 Redwood Tr, Phoenix",
      "576 Magnolia Ct, Miami"
   ],
   "Price": [625000, 789000, 425000, 550000, 920000, 385000, 675000, 445000, 510000,
830000],
   "Bedrooms": [3, 4, 2, 3, 5, 2, 3, 3, 4, 4],
   "Bathrooms": [2.0, 2.5, 1.0, 2.0, 3.5, 1.5, 2.0, 2.0, 2.5, 3.0],
   "SqFt": [1850, 2450, 1200, 1750, 3200, 1100, 1950, 1650, 2200, 2800],
   "YearBuilt": [1995, 2005, 1980, 1998, 2015, 1975, 2000, 1985, 2008, 2018],
   "Garage": [True, True, False, True, True, False, True, False, True, True],
   "WalkScore": [78, 65, 92, 70, 55, 85, 72, 68, 60, 88]
}


df = pd.DataFrame(house_price_data)


# 1. Display DataFrame info
```

```
print("1. DataFrame Info:")
print(df.info())

# 2. Show first 3 records
print("\n2. First 3 houses:")
print(df.head(3))

# 3. Calculate average price
print("\n3. Average price: ${:,.2f}".format(df['Price'].mean()))

# 4. Find most expensive house
print("\n4. Most expensive house:")
print(df[df['Price'] == df['Price'].max()][['Address', 'Price']])

# 5. Count houses with 3+ bedrooms
print("\n5. Houses with 3+ bedrooms:", (df['Bedrooms'] >= 3).sum())

# 6. Average price by bedroom count
print("\n6. Average price by bedrooms:")
print(df.groupby('Bedrooms')['Price'].mean())

# 7. Filter houses under $500K
print("\n7. Affordable houses (<$500K):")
print(df[df['Price'] < 500000][['ID', 'Address', 'Price']])

# 8. Add price per sqft column
df['PricePerSqFt'] = df['Price'] / df['SqFt']
print("\n8. Added Price/SqFt column:")
print(df[['Address', 'PricePerSqFt']].head(3))

# 9. Find newest house
```

```
print("\n9. Newest house:")
print(df[df['YearBuilt'] == df['YearBuilt'].max()][['Address', 'YearBuilt']])

# 10. Houses with garages
print("\n10. Houses with garages:", df['Garage'].sum())

# 11. Highest Walk Score
print("\n11. Highest Walk Score:", df['WalkScore'].max())

# 12. Extract cities from addresses
df['City'] = df['Address'].str.split(',').str[1].str.strip()
print("\n12. Extracted cities:")
print(df['City'].unique())

# 13. Average price by city
print("\n13. Average price by city:")
print(df.groupby('City')['Price'].mean())

# 14. Price statistics
print("\n14. Price statistics:")
print(df['Price'].describe())

# 15. Correlation matrix
print("\n15. Correlation matrix:")
print(df[['Price', 'SqFt', 'Bedrooms', 'Bathrooms']].corr())

# 16. Filter houses with 2+ baths and garage
print("\n16. Houses with 2+ baths and garage:")
print(df[(df['Bathrooms'] >= 2) & (df['Garage'])][['Address', 'Bathrooms']])

# 17. Oldest 3 houses
```

```
print("\n17. Oldest 3 houses:")
print(df.nsmallest(3, 'YearBuilt')[['Address', 'YearBuilt']])

# 18. Walk Score distribution
print("\n18. Walk Score distribution:")
print(df['WalkScore'].value_counts(bins=3))

# 19. Add age column
df['Age'] = 2023 - df['YearBuilt']
print("\n19. Added Age column:")
print(df[['Address', 'YearBuilt', 'Age']].head(3))

# 20. Save filtered data (with error handling)
try:
    df[df['YearBuilt'] > 2000].to_csv('recent_homes.csv', index=False)
    print("\n20. Saved houses built after 2000 to 'recent_homes.csv'")
except Exception as e:
    print("\n20. Error saving file:", str(e))
```

OUTPUT

```
1. DataFrame Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 9 columns):
 # Column      Non-Null Count Dtype
--- ------      -------------- -----
 0 ID          10 non-null    int64
 1 Address     10 non-null    object
 2 Price       10 non-null    int64
 3 Bedrooms 10 non-null       int64
 4 Bathrooms 10 non-null      float64
```

```
 5 SqFt        10 non-null     int64
 6 YearBuilt 10 non-null       int64
 7 Garage      10 non-null     bool
 8 WalkScore 10 non-null int64
dtypes: bool(1), float64(1), int64(6), object(1)
memory usage: 782.0+ bytes
None

2. First 3 houses:
ID                  Address Price ... YearBuilt Garage WalkScore
0 1001      123 Oak St, Seattle 625000 ...        1995    True        78
1 1002 456 Maple Ave, Portland 789000 ...          2005 True         65
2 1003      789 Pine Rd, Austin 425000 ...        1980 False         92

[3 rows x 9 columns]

3. Average price: $615,400.00

4. Most expensive house:
              Address Price
 4 654 Cedar Ln, Boston 920000

5. Houses with 3+ bedrooms: 8

6. Average price by bedrooms:
Bedrooms
2    405000.000000
3    573750.000000
4    709666.666667
5    920000.000000
Name: Price, dtype: float64
```

7. Affordable houses (<$500K):

	ID	Address	Price
2	1003	789 Pine Rd, Austin	425000
5	1006	987 Birch Dr, Atlanta	385000
7	1008	864 Willow Cir, Dallas	445000

8. Added Price/SqFt column:

	Address	PricePerSqFt
0	123 Oak St, Seattle	337.837838
1	456 Maple Ave, Portland	322.040816
2	789 Pine Rd, Austin	354.166667

9. Newest house:

	Address	YearBuilt
9	576 Magnolia Ct, Miami	2018

10. Houses with garages: 7

11. Highest Walk Score: 92

12. Extracted cities:

['Seattle' 'Portland' 'Austin' 'Denver' 'Boston' 'Atlanta' 'Chicago'
 'Dallas' 'Phoenix' 'Miami']

13. Average price by city:

City	
Atlanta	385000.0
Austin	425000.0
Boston	920000.0
Chicago	675000.0

Dallas 445000.0

Denver 550000.0

Miami 830000.0

Phoenix 510000.0

Portland 789000.0

Seattle 625000.0

Name: Price, dtype: float64

14. Price statistics:

count 10.0000

mean 615400.0000

std 184503.6585

min 385000.0000

25% 461250.0000

50% 587500.0000

75% 760500.0000

max 920000.0000

Name: Price, dtype: float64

15. Correlation matrix:

	Price	SqFt	Bedrooms	Bathrooms
Price	1.000000	0.928646	0.833355	0.851789
SqFt	0.928646	1.000000	0.969526	0.968703
Bedrooms	0.833355	0.969526	1.000000	0.966564
Bathrooms	0.851789	0.968703	0.966564	1.000000

16. Houses with 2+ baths and garage:

	Address	Bathrooms
0	123 Oak St, Seattle	2.0
1	456 Maple Ave, Portland	2.5
3	321 Elm Blvd, Denver	2.0

4 654 Cedar Ln, Boston 3.5

6 135 Spruce Way, Chicago 2.0

8 249 Redwood Tr, Phoenix 2.5

9 576 Magnolia Ct, Miami 3.0

17. Oldest 3 houses:

Address YearBuilt

5 987 Birch Dr, Atlanta 1975

2 789 Pine Rd, Austin 1980

7 864 Willow Cir, Dallas 1985

18. Walk Score distribution:

(67.333, 79.667] 4

(54.962, 67.333] 3

(79.667, 92.0] 3

Name: count, dtype: int64

19. Added Age column:

Address YearBuilt Age

0 123 Oak St, Seattle 1995 28

1 456 Maple Ave, Portland 2005 18

2 789 Pine Rd, Austin 1980 43

20. Error saving file: [Errno 13] Permission denied: 'recent_homes.csv'

=== Code Execution Successful ===