EDS THEORY ACTIVITY 1

NAME: BHAKTI KHANZODE

DIVISION: CS1

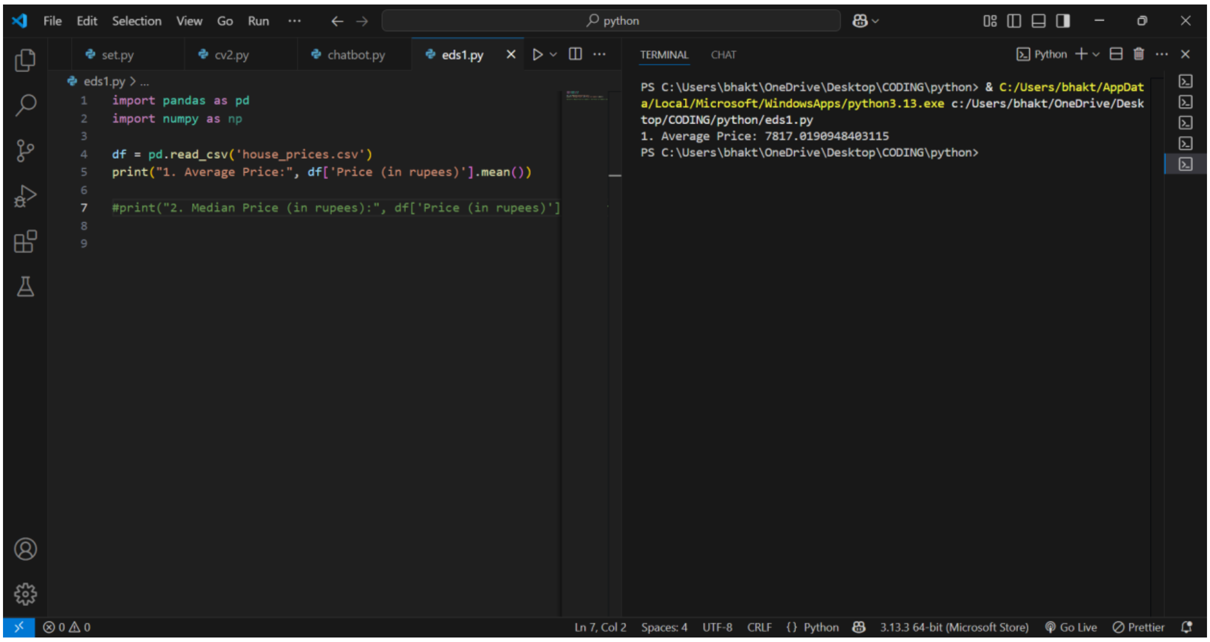
ROLL NO.: CS1-68

PRN: 202401040046

DATASET: HOUSE PRICE

PROBLEM STATEMENT:

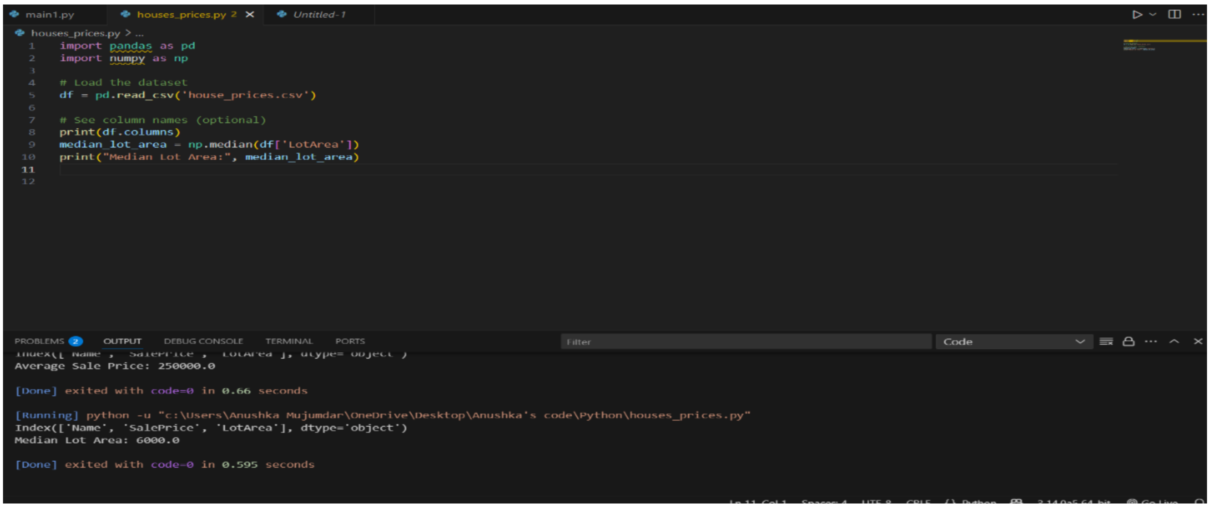
1. **Average house price**



```python
import pandas as pd
import numpy as np

df = pd.read_csv('house_prices.csv')
print("1. Average Price:", df['Price (in rupees)'].mean())

#print("2. Median Price (in rupees):", df['Price (in rupees)']
```

```
PS C:\Users\bhakt\OneDrive\Desktop\CODING\python> & C:/Users/bhakt/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/bhakt/OneDrive/Desktop/CODING/python/eds1.py
1. Average Price: 7817.0190948403115
PS C:\Users\bhakt\OneDrive\Desktop\CODING\python>
```

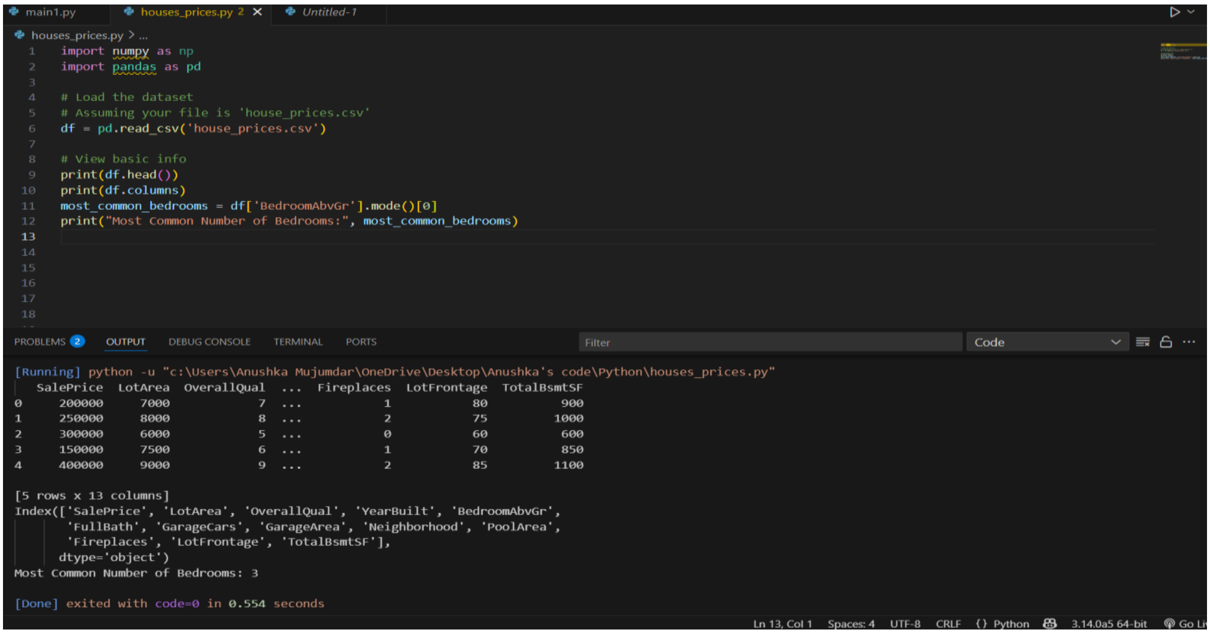
2.Find the median lot area.



```python
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('house_prices.csv')

# See column names (optional)
print(df.columns)
median_lot_area = np.median(df['LotArea'])
print("Median Lot Area:", median_lot_area)
```

```
Index(['Name', 'SalePrice', 'LotArea'], dtype='object')
Average Sale Price: 250000.0

[Done] exited with code=0 in 0.66 seconds

[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
Index(['Name', 'SalePrice', 'LotArea'], dtype='object')
Median Lot Area: 6000.0

[Done] exited with code=0 in 0.595 seconds
```

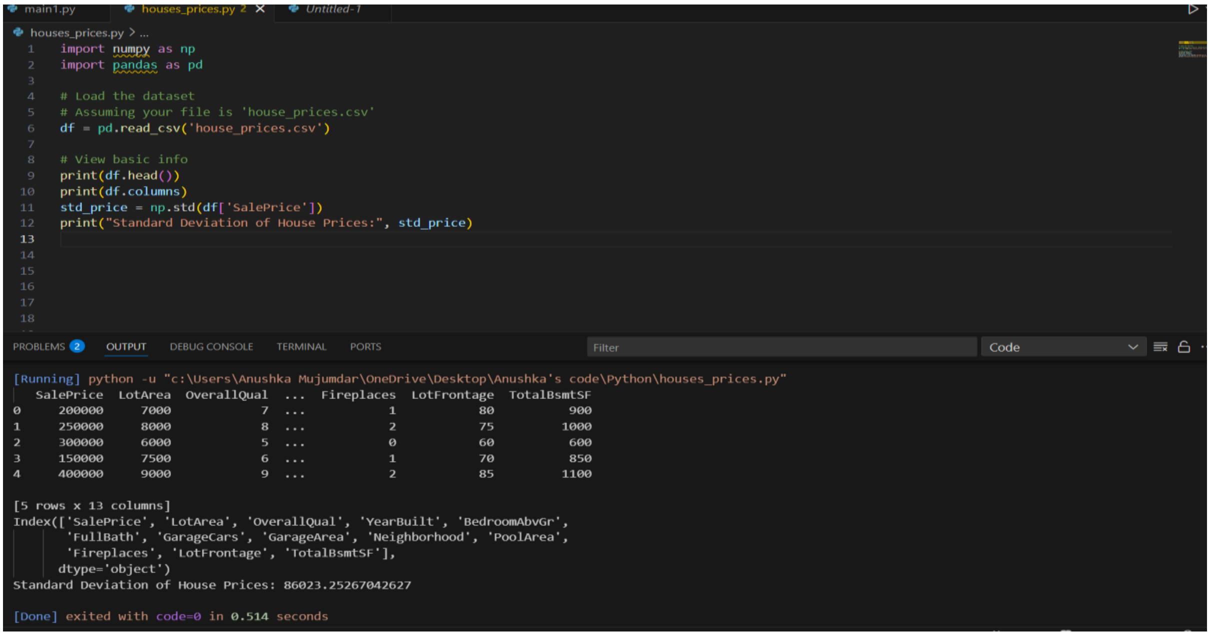
3.Find the maximum living area (GrLivArea) of the houses.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
most_common_bedrooms = df['BedroomAbvGr'].mode()[0]
print("Most Common Number of Bedrooms:", most_common_bedrooms)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Most Common Number of Bedrooms: 3

[Done] exited with code=0 in 0.554 seconds
```

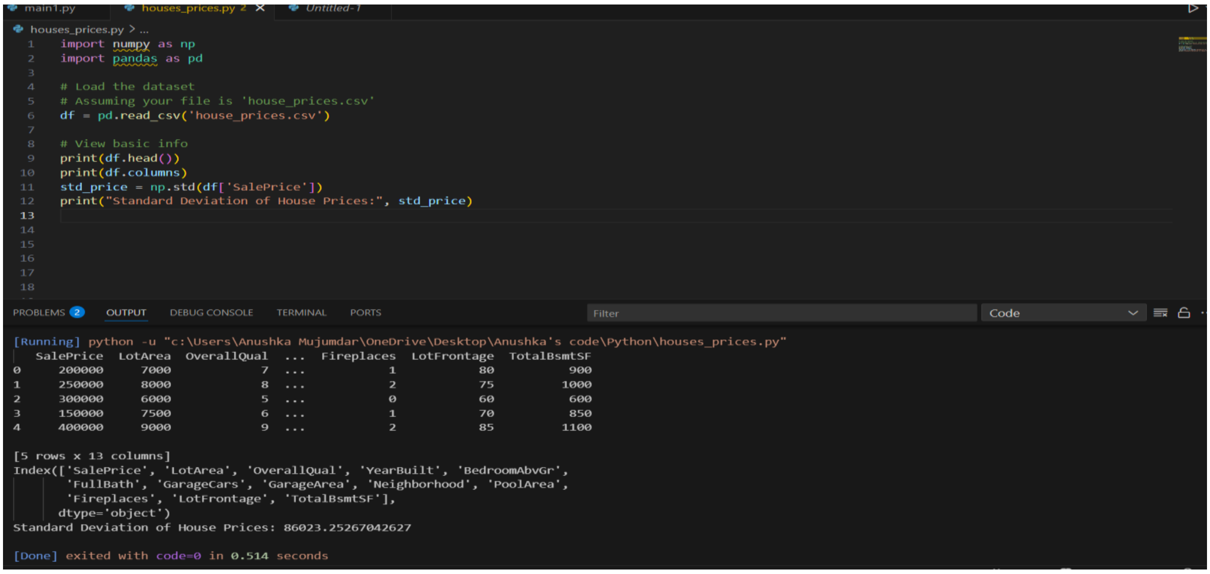
4.Find the minimum number of bedrooms.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
std_price = np.std(df['SalePrice'])
print("Standard Deviation of House Prices:", std_price)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Standard Deviation of House Prices: 86023.25267042627

[Done] exited with code=0 in 0.514 seconds
```

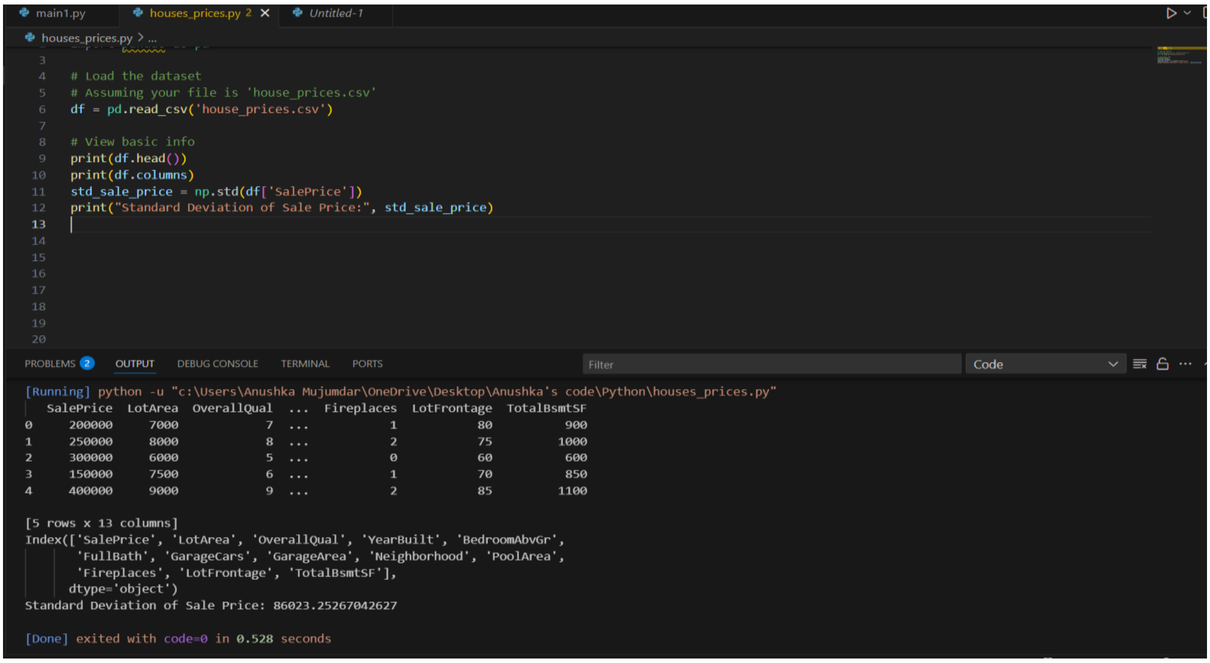
5.How many houses have more than 4 bedrooms?



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
std_price = np.std(df['SalePrice'])
print("Standard Deviation of House Prices:", std_price)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Standard Deviation of House Prices: 86023.25267042627

[Done] exited with code=0 in 0.514 seconds
```

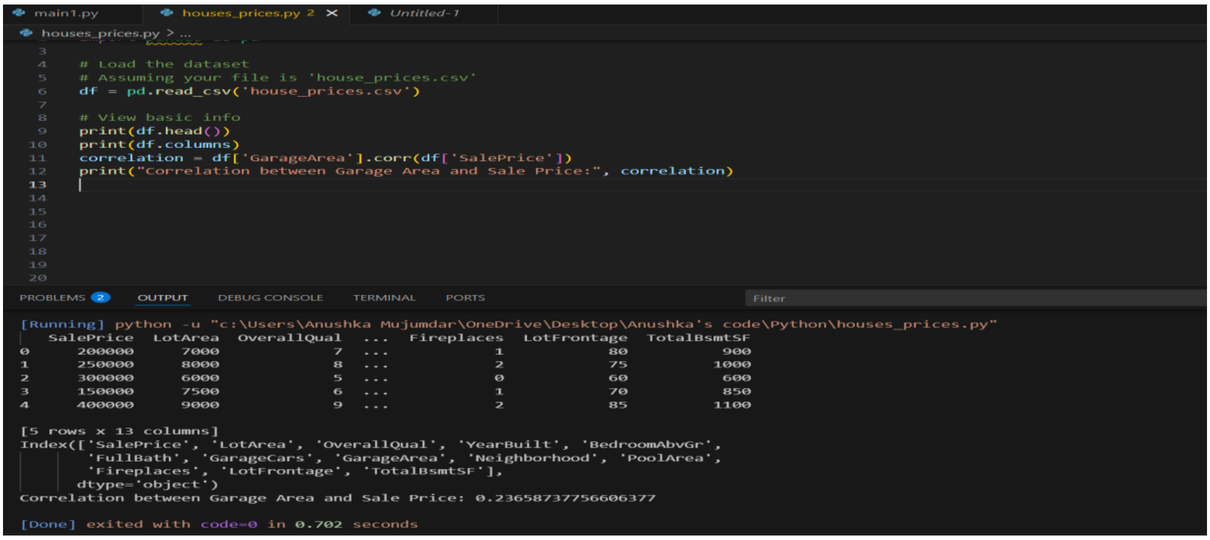
6.Find the standard deviation of house sale prices.



```
# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
std_sale_price = np.std(df['SalePrice'])
print("Standard Deviation of Sale Price:", std_sale_price)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Standard Deviation of Sale Price: 86023.25267042627

[Done] exited with code=0 in 0.528 seconds
```

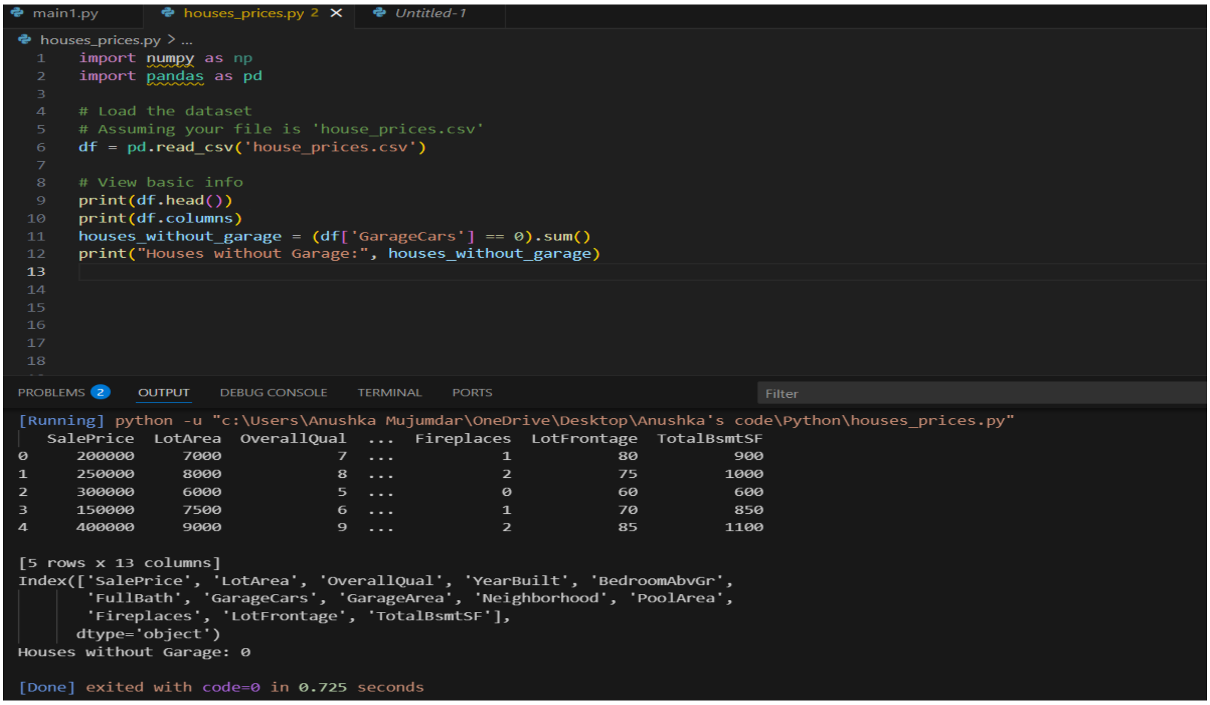
7.Find the correlation between garage area and sale price.



```
# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
correlation = df['GarageArea'].corr(df['SalePrice'])
print("Correlation between Garage Area and Sale Price:", correlation)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Correlation between Garage Area and Sale Price: 0.23658737756606377

[Done] exited with code=0 in 0.702 seconds
```

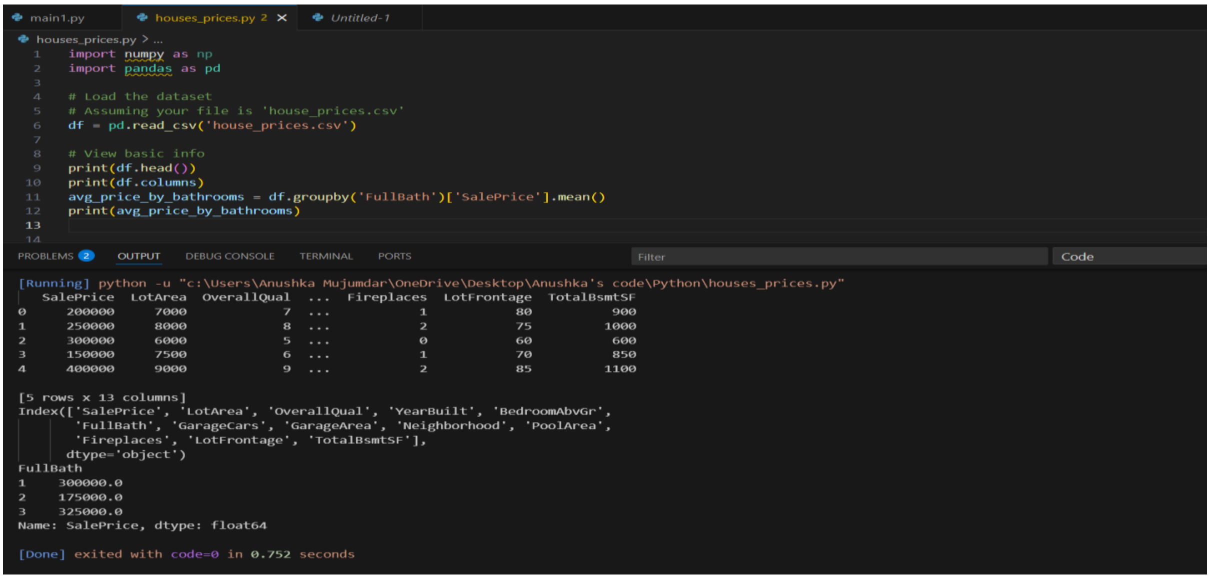
8.Find the number of houses without a garage.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
houses_without_garage = (df['GarageCars'] == 0).sum()
print("Houses without Garage:", houses_without_garage)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Houses without Garage: 0

[Done] exited with code=0 in 0.725 seconds
```

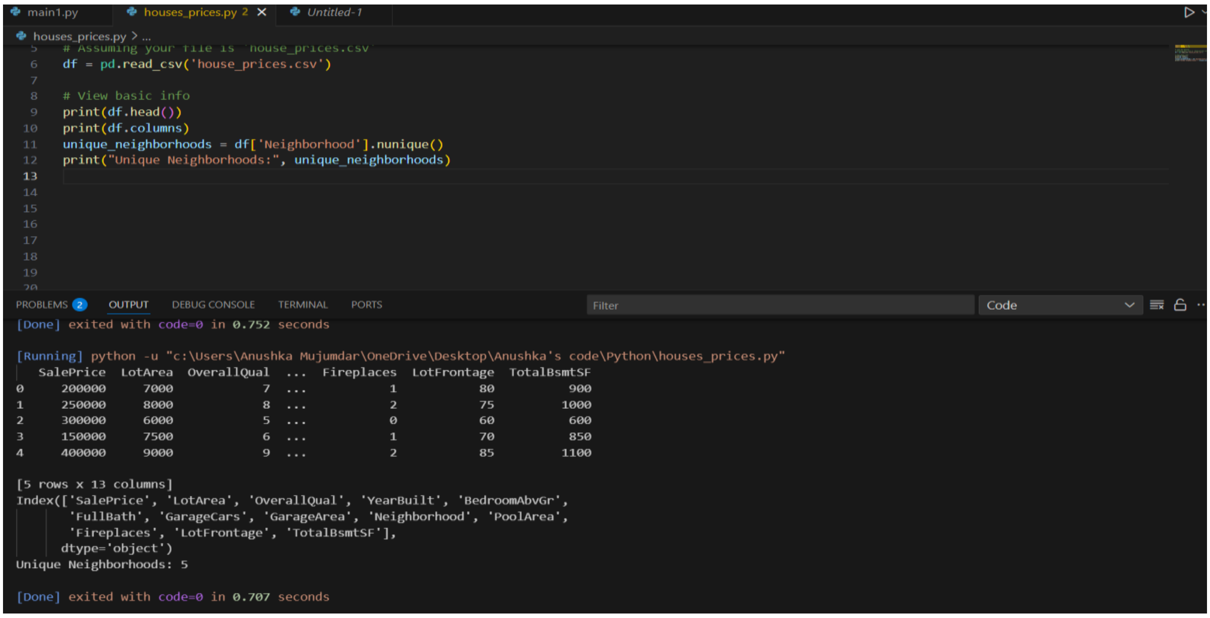
9.Group houses by number of full bathrooms and calculate the average sale price.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
avg_price_by_bathrooms = df.groupby('FullBath')['SalePrice'].mean()
print(avg_price_by_bathrooms)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
FullBath
1    300000.0
2    175000.0
3    325000.0
Name: SalePrice, dtype: float64

[Done] exited with code=0 in 0.752 seconds
```

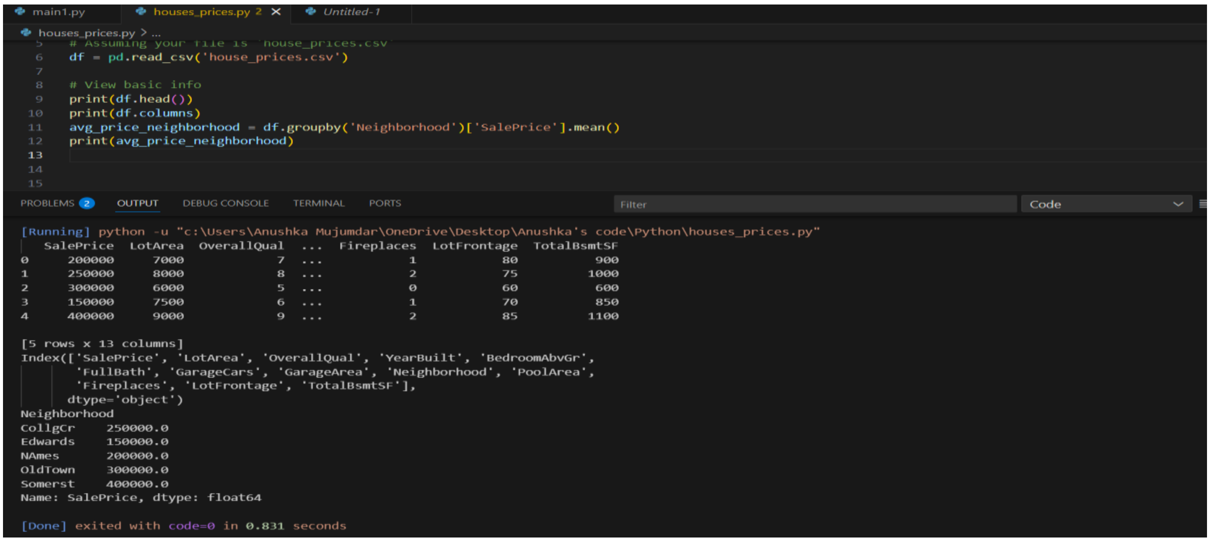
10.How many unique neighborhoods are there?



```python
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
unique_neighborhoods = df['Neighborhood'].nunique()
print("Unique Neighborhoods:", unique_neighborhoods)
```

```
[Done] exited with code=0 in 0.752 seconds

[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Unique Neighborhoods: 5

[Done] exited with code=0 in 0.707 seconds
```

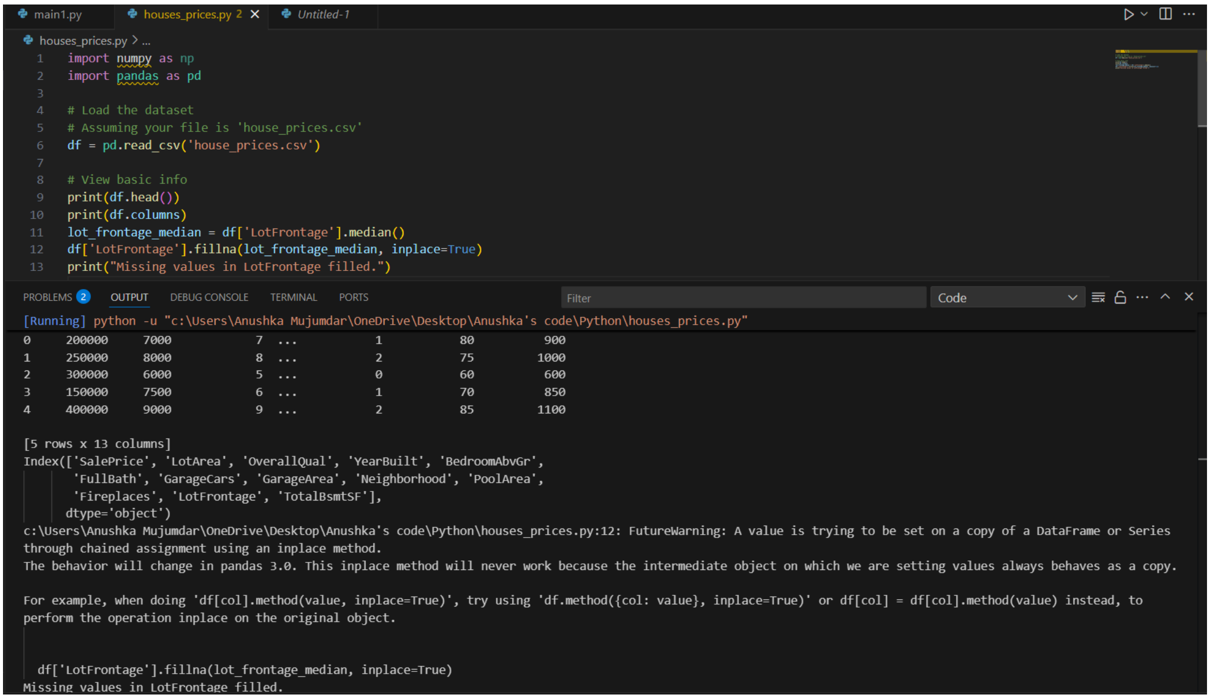
11.Find the average sale price for each neighborhood.



```python
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
avg_price_neighborhood = df.groupby('Neighborhood')['SalePrice'].mean()
print(avg_price_neighborhood)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Neighborhood
CollgCr    250000.0
Edwards    150000.0
NAmes      200000.0
OldTown    300000.0
Somerst    400000.0
Name: SalePrice, dtype: float64

[Done] exited with code=0 in 0.831 seconds
```

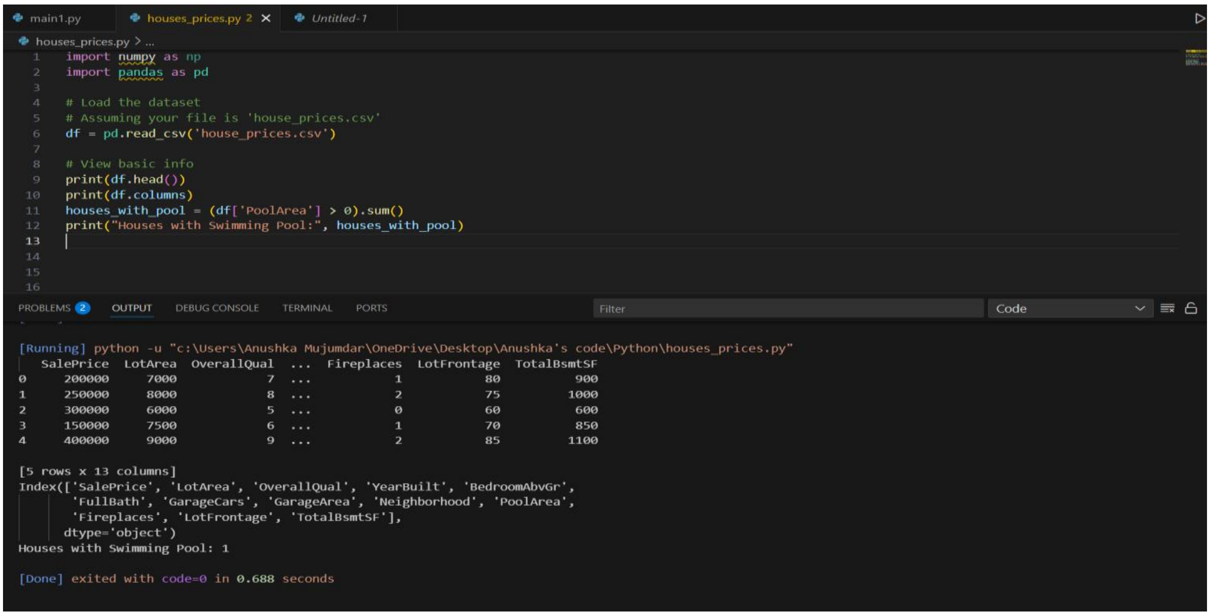
12.Fill missing values in LotFrontage with the column’s median value.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
lot_frontage_median = df['LotFrontage'].median()
df['LotFrontage'].fillna(lot_frontage_median, inplace=True)
print("Missing values in LotFrontage filled.")
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
0    200000    7000           7  ...          1          80         900
1    250000    8000           8  ...          2          75        1000
2    300000    6000           5  ...          0          60         600
3    150000    7500           6  ...          1          70         850
4    400000    9000           9  ...          2          85        1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py:12: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to
perform the operation inplace on the original object.


  df['LotFrontage'].fillna(lot_frontage_median, inplace=True)
Missing values in LotFrontage filled.
```

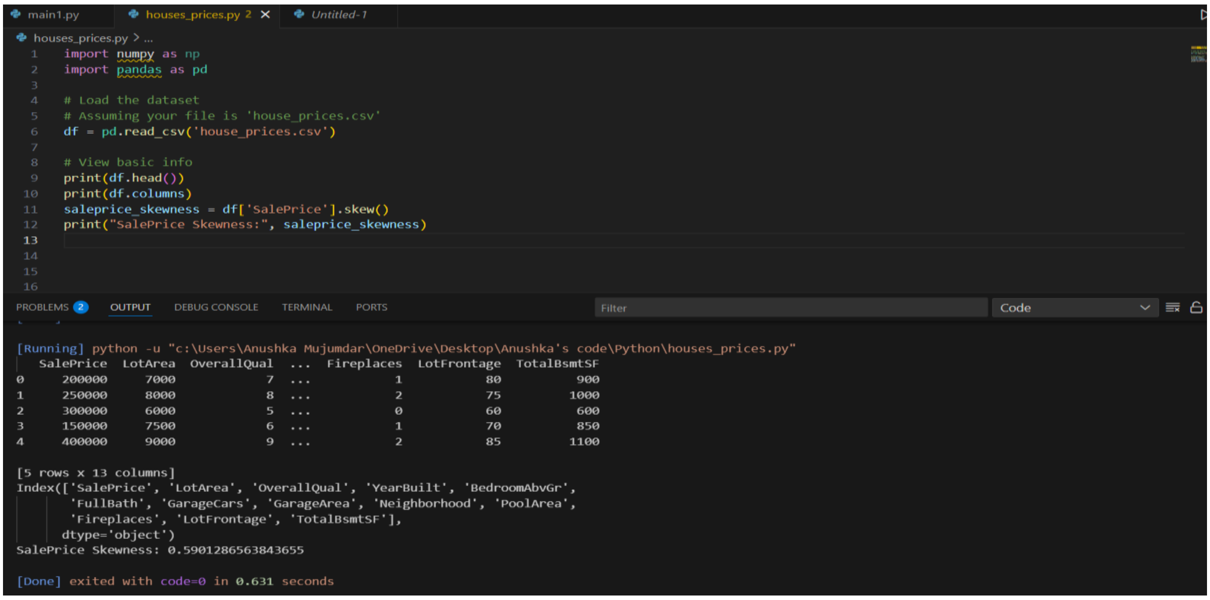
13.How many houses have a swimming pool?



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
houses_with_pool = (df['PoolArea'] > 0).sum()
print("Houses with Swimming Pool:", houses_with_pool)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Houses with Swimming Pool: 1

[Done] exited with code=0 in 0.688 seconds
```

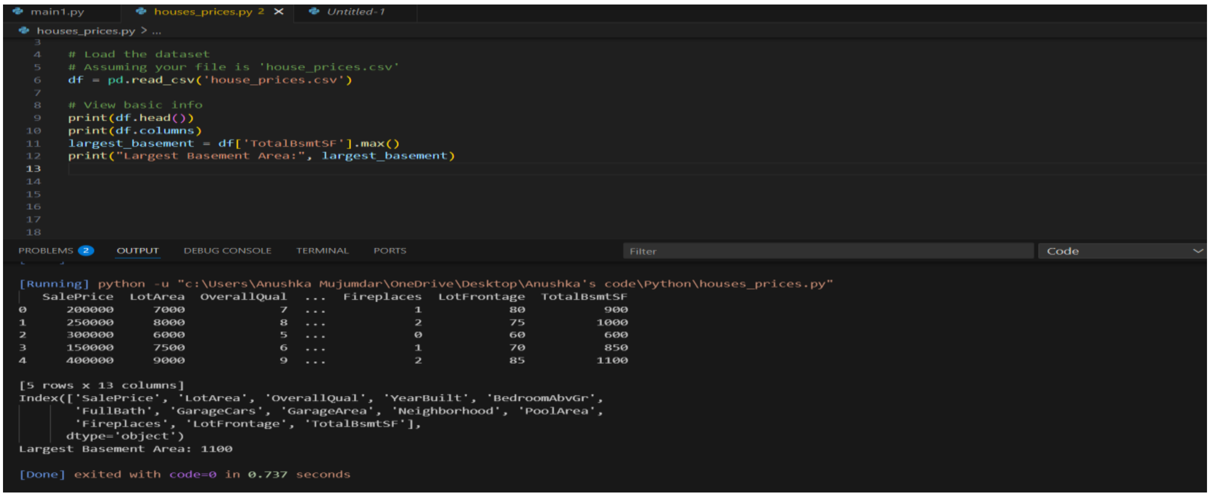
14.What is the skewness of the SalePrice?



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
saleprice_skewness = df['SalePrice'].skew()
print("SalePrice Skewness:", saleprice_skewness)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
SalePrice Skewness: 0.5901286563843655

[Done] exited with code=0 in 0.631 seconds
```

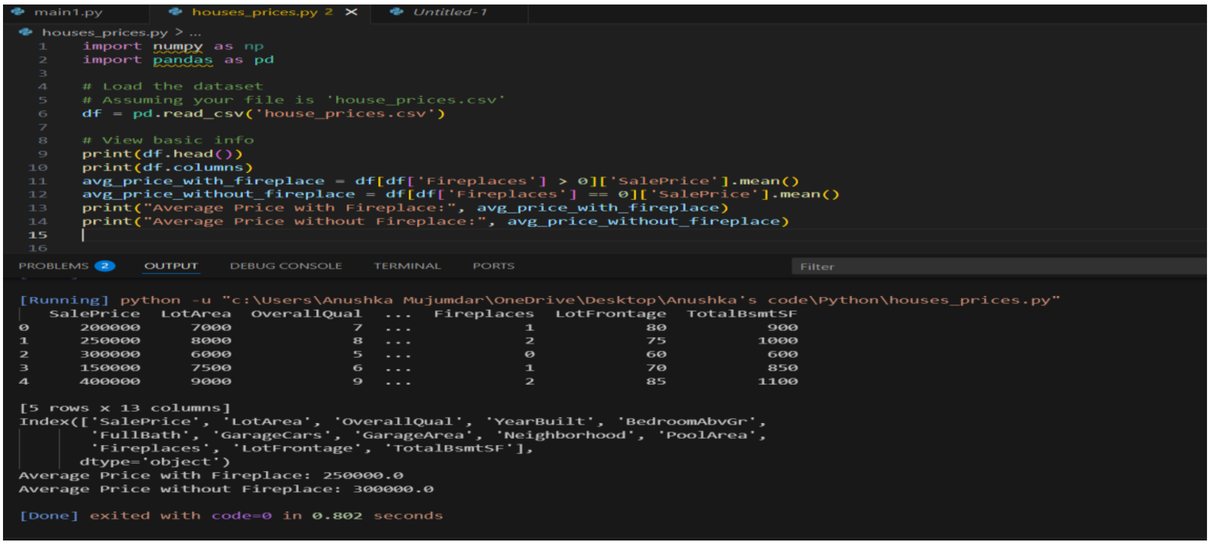
15.Find the house with the largest total basement area (TotalBsmtSF).



```python

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
largest_basement = df['TotalBsmtSF'].max()
print("Largest Basement Area:", largest_basement)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Largest Basement Area: 1100

[Done] exited with code=0 in 0.737 seconds
```

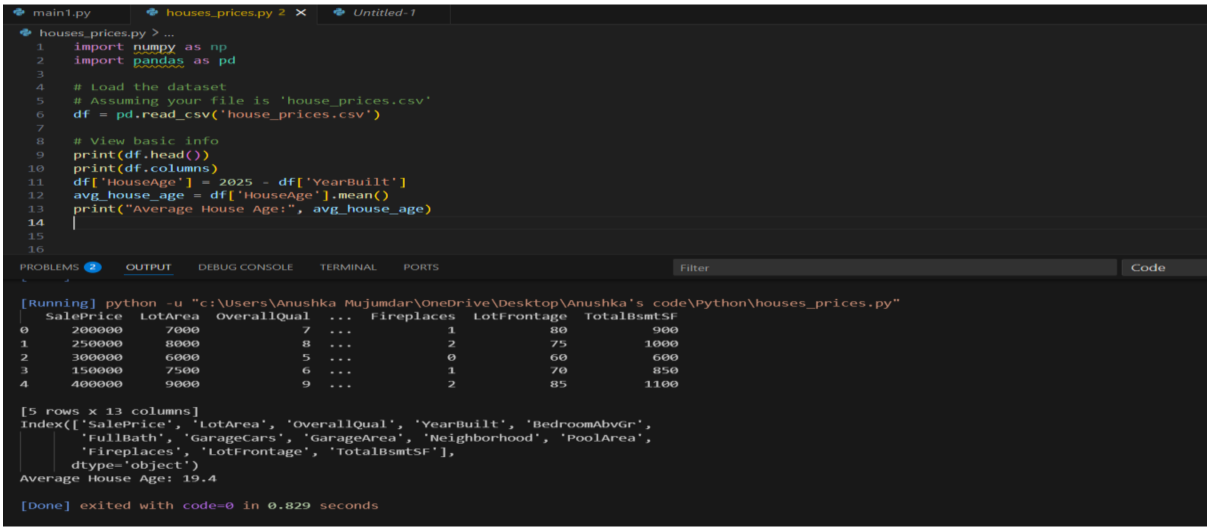
16.Find the average sale price for houses with and without fireplaces.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
avg_price_with_fireplace = df[df['Fireplaces'] > 0]['SalePrice'].mean()
avg_price_without_fireplace = df[df['Fireplaces'] == 0]['SalePrice'].mean()
print("Average Price with Fireplace:", avg_price_with_fireplace)
print("Average Price without Fireplace:", avg_price_without_fireplace)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Average Price with Fireplace: 250000.0
Average Price without Fireplace: 300000.0

[Done] exited with code=0 in 0.802 seconds
```

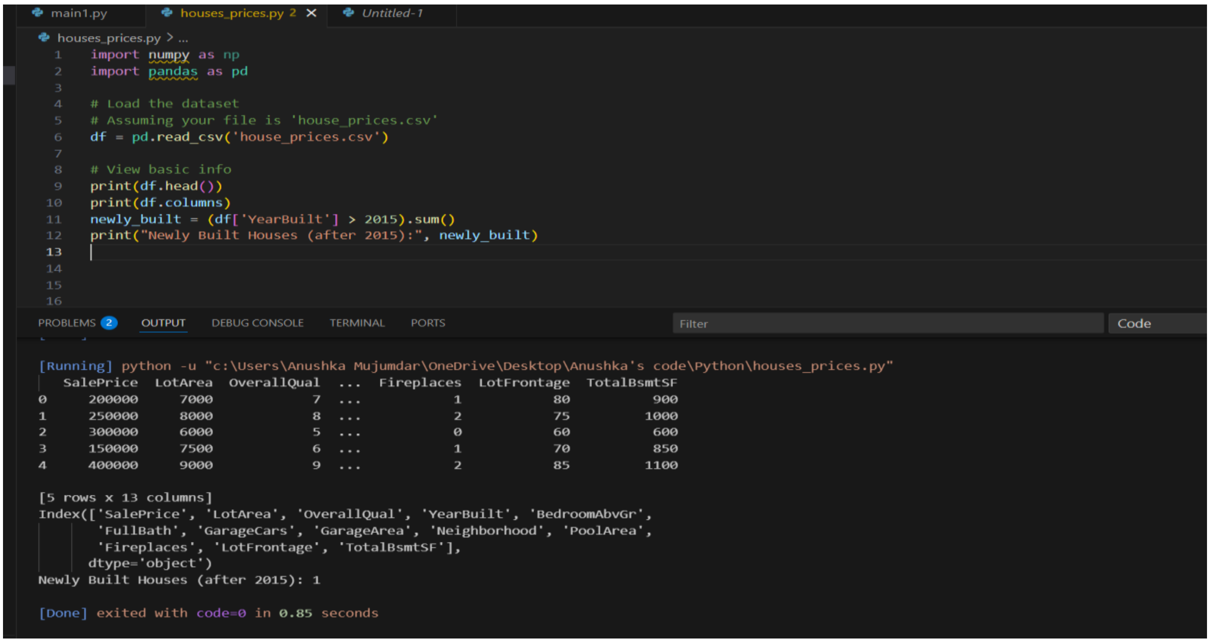
17.Create a new column for the house age and find the average age.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
df['HouseAge'] = 2025 - df['YearBuilt']
avg_house_age = df['HouseAge'].mean()
print("Average House Age:", avg_house_age)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Average House Age: 19.4

[Done] exited with code=0 in 0.829 seconds
```

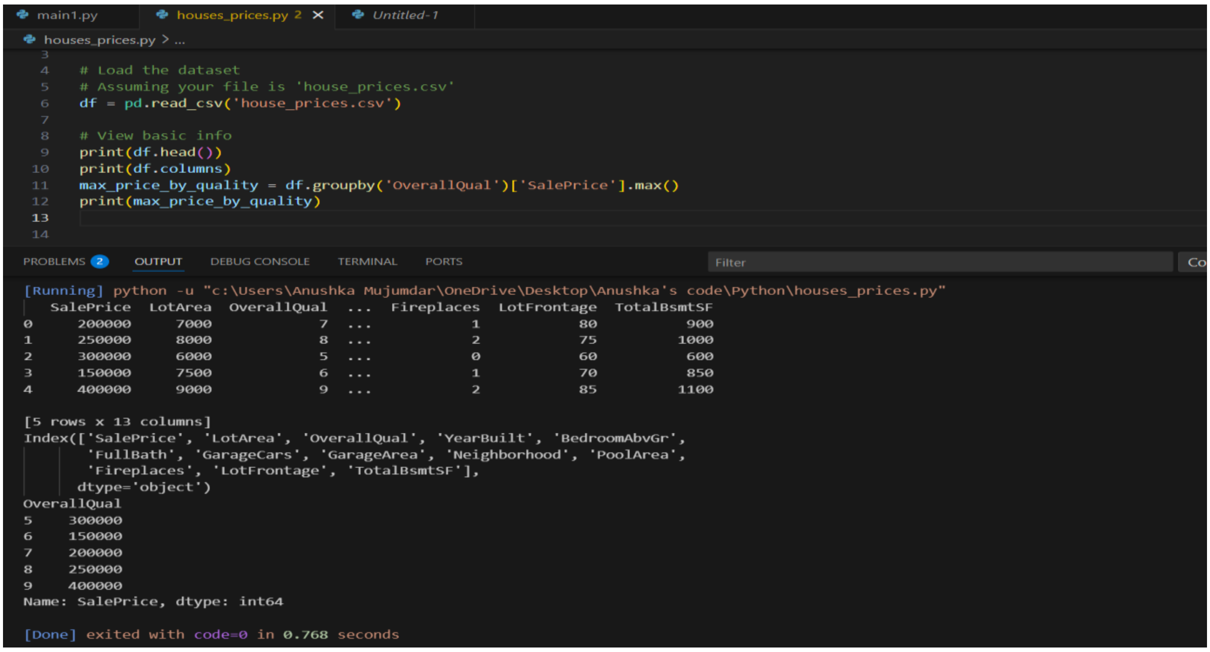
18.How many houses are newly built (after 2015)?



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
newly_built = (df['YearBuilt'] > 2015).sum()
print("Newly Built Houses (after 2015):", newly_built)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Newly Built Houses (after 2015): 1

[Done] exited with code=0 in 0.85 seconds
```

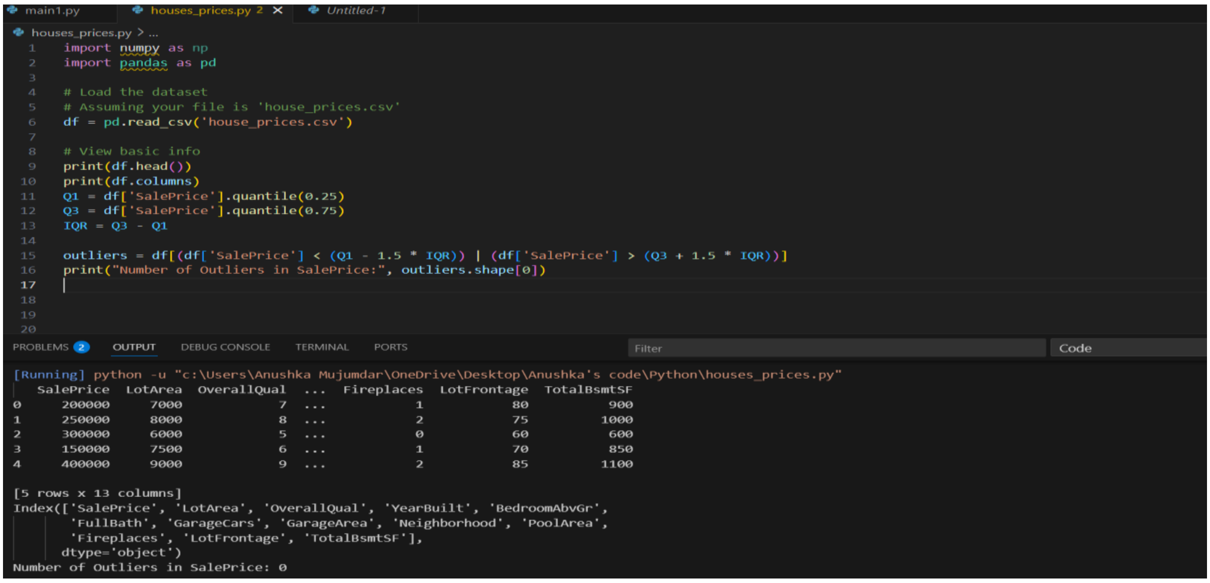
19. Group houses by overall quality (OverallQual) and find maximum sale price in each group.



```python

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
max_price_by_quality = df.groupby('OverallQual')['SalePrice'].max()
print(max_price_by_quality)
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
OverallQual
5    300000
6    150000
7    200000
8    250000
9    400000
Name: SalePrice, dtype: int64

[Done] exited with code=0 in 0.768 seconds
```

20. Find outliers in SalePrice using the IQR method.



```python
import numpy as np
import pandas as pd

# Load the dataset
# Assuming your file is 'house_prices.csv'
df = pd.read_csv('house_prices.csv')

# View basic info
print(df.head())
print(df.columns)
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))]
print("Number of Outliers in SalePrice:", outliers.shape[0])
```

```
[Running] python -u "c:\Users\Anushka Mujumdar\OneDrive\Desktop\Anushka's code\Python\houses_prices.py"
   SalePrice  LotArea  OverallQual  ...  Fireplaces  LotFrontage  TotalBsmtSF
0     200000     7000            7  ...           1           80          900
1     250000     8000            8  ...           2           75         1000
2     300000     6000            5  ...           0           60          600
3     150000     7500            6  ...           1           70          850
4     400000     9000            9  ...           2           85         1100

[5 rows x 13 columns]
Index(['SalePrice', 'LotArea', 'OverallQual', 'YearBuilt', 'BedroomAbvGr',
       'FullBath', 'GarageCars', 'GarageArea', 'Neighborhood', 'PoolArea',
       'Fireplaces', 'LotFrontage', 'TotalBsmtSF'],
      dtype='object')
Number of Outliers in SalePrice: 0
```