

assignment1

April 26, 2024

0.1 Assignment 1

Name - Jay Shirgupe

PRN - 21510026

Batch - T7

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
```

```
[ ]: df = pd.read_csv('home_data.csv')
```

```
[ ]: df.head()
```

```
[ ]:
      id          date  price  bedrooms  bathrooms  sqft_living \
0  7129300520  20141013T000000  221900         3         1.00        1180
1  6414100192  20141209T000000  538000         3         2.25        2570
2  5631500400  20150225T000000  180000         2         1.00         770
3  2487200875  20141209T000000  604000         4         3.00        1960
4  1954400510  20150218T000000  510000         3         2.00        1680

      sqft_lot  floors  waterfront  view  ...  grade  sqft_above  sqft_basement \
0         5650     1.0           0     0  ...     7         1180           0
1         7242     2.0           0     0  ...     7         2170          400
2        10000     1.0           0     0  ...     6          770           0
3         5000     1.0           0     0  ...     7         1050          910
4         8080     1.0           0     0  ...     8         1680           0

      yr_built  yr_renovated  zipcode      lat      long  sqft_living15 \
0         1955           0    98178  47.5112 -122.257        1340
1         1951        1991    98125  47.7210 -122.319        1690
2         1933           0    98028  47.7379 -122.233        2720
3         1965           0    98136  47.5208 -122.393        1360
4         1987           0    98074  47.6168 -122.045        1800

      sqft_lot15
0         5650
1         7639
```

```
2      8062
3      5000
4      7503
```

[5 rows x 21 columns]

1. Display shape of data frame

```
[ ]: df.shape
```

```
[ ]: (21613, 21)
```

2. Display column names

```
[ ]: df.columns
```

```
[ ]: Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
           'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
           'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
           'lat', 'long', 'sqft_living15', 'sqft_lot15'],
          dtype='object')
```

3. Display 5 quantiles of the dataset

```
[ ]: df.quantile([0, .25, .5, .75, 1], numeric_only=True)
```

```
[ ]:
```

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	\
0.00	1.000102e+06	75000.0	0.0	0.00	290.0	520.0	
0.25	2.123049e+09	321950.0	3.0	1.75	1427.0	5040.0	
0.50	3.904930e+09	450000.0	3.0	2.25	1910.0	7618.0	
0.75	7.308900e+09	645000.0	4.0	2.50	2550.0	10688.0	
1.00	9.900000e+09	7700000.0	33.0	8.00	13540.0	1651359.0	

	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	\
0.00	1.0	0.0	0.0	1.0	1.0	290.0	0.0	
0.25	1.0	0.0	0.0	3.0	7.0	1190.0	0.0	
0.50	1.5	0.0	0.0	3.0	7.0	1560.0	0.0	
0.75	2.0	0.0	0.0	4.0	8.0	2210.0	560.0	
1.00	3.5	1.0	4.0	5.0	13.0	9410.0	4820.0	

	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	\
0.00	1900.0	0.0	98001.0	47.1559	-122.519	399.0	
0.25	1951.0	0.0	98033.0	47.4710	-122.328	1490.0	
0.50	1975.0	0.0	98065.0	47.5718	-122.230	1840.0	
0.75	1997.0	0.0	98118.0	47.6780	-122.125	2360.0	
1.00	2015.0	2015.0	98199.0	47.7776	-121.315	6210.0	

	sqft_lot15
--	------------

```

0.00      651.0
0.25     5100.0
0.50     7620.0
0.75    10083.0
1.00    871200.0

```

4. Display count of rows having null in any column

```
[ ]: df.isnull()
```

```
[ ]:
      id  date  price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0   False False  False      False      False      False   False
1   False False  False      False      False      False   False
2   False False  False      False      False      False   False
3   False False  False      False      False      False   False
4   False False  False      False      False      False   False
...
21608 False False  False      False      False      False   False
21609 False False  False      False      False      False   False
21610 False False  False      False      False      False   False
21611 False False  False      False      False      False   False
21612 False False  False      False      False      False   False

```

```

      floors  waterfront  view  ...  grade  sqft_above  sqft_basement  \
0   False      False  False  ...  False      False      False
1   False      False  False  ...  False      False      False
2   False      False  False  ...  False      False      False
3   False      False  False  ...  False      False      False
4   False      False  False  ...  False      False      False
...
21608 False      False  False  ...  False      False      False
21609 False      False  False  ...  False      False      False
21610 False      False  False  ...  False      False      False
21611 False      False  False  ...  False      False      False
21612 False      False  False  ...  False      False      False

```

```

      yr_built  yr_renovated  zipcode  lat  long  sqft_living15  \
0   False      False      False  False  False      False
1   False      False      False  False  False      False
2   False      False      False  False  False      False
3   False      False      False  False  False      False
4   False      False      False  False  False      False
...
21608 False      False      False  False  False      False
21609 False      False      False  False  False      False
21610 False      False      False  False  False      False
21611 False      False      False  False  False      False

```

```
21612      False      False      False      False      False      False
```

```

      sqft_lot15
0          False
1          False
2          False
3          False
4          False
...          ...
21608      False
21609      False
21610      False
21611      False
21612      False

```

```
[21613 rows x 21 columns]
```

```
[ ]: df.isnull().any(axis=1).sum()
```

```
[ ]: 0
```

5. Display first 10 rows

```
[ ]: df.head(10)
```

```
[ ]:
      id      date      price  bedrooms  bathrooms  sqft_living  \
0  7129300520  20141013T0000000  221900          3          1.00          1180
1  6414100192  20141209T0000000  538000          3          2.25          2570
2  5631500400  20150225T0000000  180000          2          1.00          770
3  2487200875  20141209T0000000  604000          4          3.00          1960
4  1954400510  20150218T0000000  510000          3          2.00          1680
5  7237550310  20140512T0000000  1225000         4          4.50          5420
6  1321400060  20140627T0000000  257500          3          2.25          1715
7  2008000270  20150115T0000000  291850          3          1.50          1060
8  2414600126  20150415T0000000  229500          3          1.00          1780
9  3793500160  20150312T0000000  323000          3          2.50          1890

      sqft_lot  floors  waterfront  view  ...  grade  sqft_above  sqft_basement  \
0         5650     1.0           0     0  ...     7         1180           0
1         7242     2.0           0     0  ...     7         2170          400
2        10000     1.0           0     0  ...     6          770           0
3         5000     1.0           0     0  ...     7        1050          910
4         8080     1.0           0     0  ...     8        1680           0
5        101930     1.0           0     0  ...    11        3890         1530
6         6819     2.0           0     0  ...     7        1715           0
7         9711     1.0           0     0  ...     7        1060           0
8         7470     1.0           0     0  ...     7        1050          730

```

```
9      6560      2.0      0      0 ...      7      1890      0
```

```

      yr_built yr_renovated zipcode      lat      long sqft_living15 \
0      1955      0      98178 47.5112 -122.257      1340
1      1951      1991      98125 47.7210 -122.319      1690
2      1933      0      98028 47.7379 -122.233      2720
3      1965      0      98136 47.5208 -122.393      1360
4      1987      0      98074 47.6168 -122.045      1800
5      2001      0      98053 47.6561 -122.005      4760
6      1995      0      98003 47.3097 -122.327      2238
7      1963      0      98198 47.4095 -122.315      1650
8      1960      0      98146 47.5123 -122.337      1780
9      2003      0      98038 47.3684 -122.031      2390

```

```

      sqft_lot15
0      5650
1      7639
2      8062
3      5000
4      7503
5      101930
6      6819
7      9711
8      8113
9      7570

```

[10 rows x 21 columns]

6. Display last 10 rows

```
[ ]: df.tail(10)
```

```

[ ]:
      id      date      price bedrooms bathrooms sqft_living \
21603 7852140040 20140825T000000 507250      3      2.50      2270
21604 9834201367 20150126T000000 429000      3      2.00      1490
21605 3448900210 20141014T000000 610685      4      2.50      2520
21606 7936000429 20150326T000000 1007500      4      3.50      3510
21607 2997800021 20150219T000000 475000      3      2.50      1310
21608 263000018 20140521T000000 360000      3      2.50      1530
21609 6600060120 20150223T000000 400000      4      2.50      2310
21610 1523300141 20140623T000000 402101      2      0.75      1020
21611 291310100 20150116T000000 400000      3      2.50      1600
21612 1523300157 20141015T000000 325000      2      0.75      1020

      sqft_lot floors waterfront view ... grade sqft_above \
21603      5536      2.0      0      0 ...      8      2270
21604      1126      3.0      0      0 ...      8      1490

```

21605	6023	2.0	0	0	...	9	2520
21606	7200	2.0	0	0	...	9	2600
21607	1294	2.0	0	0	...	8	1180
21608	1131	3.0	0	0	...	8	1530
21609	5813	2.0	0	0	...	8	2310
21610	1350	2.0	0	0	...	7	1020
21611	2388	2.0	0	0	...	8	1600
21612	1076	2.0	0	0	...	7	1020

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
21603	0	2003	0	98065	47.5389	-121.881	
21604	0	2014	0	98144	47.5699	-122.288	
21605	0	2014	0	98056	47.5137	-122.167	
21606	910	2009	0	98136	47.5537	-122.398	
21607	130	2008	0	98116	47.5773	-122.409	
21608	0	2009	0	98103	47.6993	-122.346	
21609	0	2014	0	98146	47.5107	-122.362	
21610	0	2009	0	98144	47.5944	-122.299	
21611	0	2004	0	98027	47.5345	-122.069	
21612	0	2008	0	98144	47.5941	-122.299	

	sqft_living15	sqft_lot15
21603	2270	5731
21604	1400	1230
21605	2520	6023
21606	2050	6200
21607	1330	1265
21608	1530	1509
21609	1830	7200
21610	1020	2007
21611	1410	1287
21612	1020	1357

[10 rows x 21 columns]

7. Display rows having number of floors more than or equal to 1.5

```
[ ]: df[df['floors'] >= 1.5]['id'].count()
```

```
[ ]: 10933
```

```
[ ]: df[df['floors'] >= 1.5]
```

	id	date	price	bedrooms	bathrooms	sqft_living	\
1	6414100192	20141209T000000	538000	3	2.25	2570	
6	1321400060	20140627T000000	257500	3	2.25	1715	
9	3793500160	20150312T000000	323000	3	2.50	1890	

12	114101516	20140528T000000	310000	3	1.00	1430
14	1175000570	20150312T000000	530000	5	2.00	1810
...
21608	263000018	20140521T000000	360000	3	2.50	1530
21609	6600060120	20150223T000000	400000	4	2.50	2310
21610	1523300141	20140623T000000	402101	2	0.75	1020
21611	291310100	20150116T000000	400000	3	2.50	1600
21612	1523300157	20141015T000000	325000	2	0.75	1020

	sqft_lot	floors	waterfront	view	...	grade	sqft_above	\
1	7242	2.0	0	0	...	7	2170	
6	6819	2.0	0	0	...	7	1715	
9	6560	2.0	0	0	...	7	1890	
12	19901	1.5	0	0	...	7	1430	
14	4850	1.5	0	0	...	7	1810	
...
21608	1131	3.0	0	0	...	8	1530	
21609	5813	2.0	0	0	...	8	2310	
21610	1350	2.0	0	0	...	7	1020	
21611	2388	2.0	0	0	...	8	1600	
21612	1076	2.0	0	0	...	7	1020	

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
1	400	1951	1991	98125	47.7210	-122.319	
6	0	1995	0	98003	47.3097	-122.327	
9	0	2003	0	98038	47.3684	-122.031	
12	0	1927	0	98028	47.7558	-122.229	
14	0	1900	0	98107	47.6700	-122.394	
...
21608	0	2009	0	98103	47.6993	-122.346	
21609	0	2014	0	98146	47.5107	-122.362	
21610	0	2009	0	98144	47.5944	-122.299	
21611	0	2004	0	98027	47.5345	-122.069	
21612	0	2008	0	98144	47.5941	-122.299	

	sqft_living15	sqft_lot15
1	1690	7639
6	2238	6819
9	2390	7570
12	1780	12697
14	1360	4850
...
21608	1530	1509
21609	1830	7200
21610	1020	2007
21611	1410	1287
21612	1020	1357

[10933 rows x 21 columns]

8. Display houses older than 1968 and their count

```
[ ]: (df['yr_built'] < 1968).sum()
```

```
[ ]: 9388
```

```
[ ]: df[df['yr_built'] < 1968]
```

```
[ ]:
```

	id	date	price	bedrooms	bathrooms	sqft_living	\
0	7129300520	20141013T000000	221900	3	1.00	1180	
1	6414100192	20141209T000000	538000	3	2.25	2570	
2	5631500400	20150225T000000	180000	2	1.00	770	
3	2487200875	20141209T000000	604000	4	3.00	1960	
7	2008000270	20150115T000000	291850	3	1.50	1060	
...	
21303	522059130	20150429T000000	465000	3	1.00	1150	
21332	9266700190	20150511T000000	245000	1	1.00	390	
21343	2924079034	20140925T000000	332220	3	1.50	2580	
21470	121039156	20150109T000000	249000	3	1.00	1030	
21583	2025049203	20140610T000000	399950	2	1.00	710	

	sqft_lot	floors	waterfront	view	...	grade	sqft_above	\
0	5650	1.0	0	0	...	7	1180	
1	7242	2.0	0	0	...	7	2170	
2	10000	1.0	0	0	...	6	770	
3	5000	1.0	0	0	...	7	1050	
7	9711	1.0	0	0	...	7	1060	
...	
21303	18200	1.0	0	0	...	7	1150	
21332	2000	1.0	0	0	...	6	390	
21343	47480	1.0	0	0	...	7	1360	
21470	24750	1.0	0	2	...	5	1030	
21583	1157	2.0	0	0	...	7	710	

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
0	0	1955	0	98178	47.5112	-122.257	
1	400	1951	1991	98125	47.7210	-122.319	
2	0	1933	0	98028	47.7379	-122.233	
3	910	1965	0	98136	47.5208	-122.393	
7	0	1963	0	98198	47.4095	-122.315	
...	
21303	0	1959	0	98058	47.4262	-122.187	
21332	0	1920	0	98103	47.6938	-122.347	
21343	1220	1953	0	98024	47.5333	-121.933	

21470	0	1943	0	98023	47.3343	-122.362
21583	0	1943	0	98102	47.6413	-122.329

	sqft_living15	sqft_lot15
0	1340	5650
1	1690	7639
2	2720	8062
3	1360	5000
7	1650	9711
...
21303	1714	18200
21332	1340	5100
21343	1760	48181
21470	2810	28800
21583	1370	1173

[9388 rows x 21 columns]

9. Display houses having price more than 50000/- and their count

```
[ ]: (df['price'] > 50000)
```

```
[ ]: 0      True
      1      True
      2      True
      3      True
      4      True
      ...
      21608   True
      21609   True
      21610   True
      21611   True
      21612   True
      Name: price, Length: 21613, dtype: bool
```

```
[ ]: df[df['price'] > 50000]
```

```
[ ]:      id      date  price  bedrooms  bathrooms  sqft_living  \
0    7129300520  20141013T000000  221900         3         1.00        1180
1    6414100192  20141209T000000  538000         3         2.25        2570
2    5631500400  20150225T000000  180000         2         1.00         770
3    2487200875  20141209T000000  604000         4         3.00        1960
4    1954400510  20150218T000000  510000         3         2.00        1680
...  ...  ...  ...  ...  ...  ...
21608  263000018  20140521T000000  360000         3         2.50        1530
21609  6600060120  20150223T000000  400000         4         2.50        2310
21610  1523300141  20140623T000000  402101         2         0.75        1020
```

21611	291310100	20150116T000000	400000	3	2.50	1600
21612	1523300157	20141015T000000	325000	2	0.75	1020

	sqft_lot	floors	waterfront	view	...	grade	sqft_above	\
0	5650	1.0	0	0	...	7	1180	
1	7242	2.0	0	0	...	7	2170	
2	10000	1.0	0	0	...	6	770	
3	5000	1.0	0	0	...	7	1050	
4	8080	1.0	0	0	...	8	1680	
...	
21608	1131	3.0	0	0	...	8	1530	
21609	5813	2.0	0	0	...	8	2310	
21610	1350	2.0	0	0	...	7	1020	
21611	2388	2.0	0	0	...	8	1600	
21612	1076	2.0	0	0	...	7	1020	

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
0	0	1955	0	98178	47.5112	-122.257	
1	400	1951	1991	98125	47.7210	-122.319	
2	0	1933	0	98028	47.7379	-122.233	
3	910	1965	0	98136	47.5208	-122.393	
4	0	1987	0	98074	47.6168	-122.045	
...	
21608	0	2009	0	98103	47.6993	-122.346	
21609	0	2014	0	98146	47.5107	-122.362	
21610	0	2009	0	98144	47.5944	-122.299	
21611	0	2004	0	98027	47.5345	-122.069	
21612	0	2008	0	98144	47.5941	-122.299	

	sqft_living15	sqft_lot15
0	1340	5650
1	1690	7639
2	2720	8062
3	1360	5000
4	1800	7503
...
21608	1530	1509
21609	1830	7200
21610	1020	2007
21611	1410	1287
21612	1020	1357

[21613 rows x 21 columns]

- Display houses having number of bedrooms less than or equal to 2 and price above 51000/- along with their count

```
[ ]: selected_houses = df[df['price'] > 51000][df['bedrooms'] <= 2]
```

```
[ ]: selected_houses
```

```
[ ]:
```

	id	date	price	bedrooms	bathrooms	sqft_living	\
2	5631500400	20150225T000000	180000	2	1.00	770	
11	9212900260	20140527T000000	468000	2	1.00	1160	
18	16000397	20141205T000000	189000	2	1.00	1200	
23	8091400200	20140516T000000	252700	2	1.50	1070	
31	2426039314	20141201T000000	280000	2	1.50	1190	
...	
21586	2767604724	20141015T000000	505000	2	2.50	1430	
21588	2767600688	20141113T000000	414500	2	1.50	1210	
21595	1972201967	20141031T000000	520000	2	2.25	1530	
21610	1523300141	20140623T000000	402101	2	0.75	1020	
21612	1523300157	20141015T000000	325000	2	0.75	1020	

	sqft_lot	floors	waterfront	view	...	grade	sqft_above	\
2	10000	1.0	0	0	...	6	770	
11	6000	1.0	0	0	...	7	860	
18	9850	1.0	0	0	...	7	1200	
23	9643	1.0	0	0	...	7	1070	
31	1265	3.0	0	0	...	7	1190	
...	
21586	1201	3.0	0	0	...	8	1430	
21588	1278	2.0	0	0	...	8	1020	
21595	981	3.0	0	0	...	8	1480	
21610	1350	2.0	0	0	...	7	1020	
21612	1076	2.0	0	0	...	7	1020	

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
2	0	1933	0	98028	47.7379	-122.233	
11	300	1942	0	98115	47.6900	-122.292	
18	0	1921	0	98002	47.3089	-122.210	
23	0	1985	0	98030	47.3533	-122.166	
31	0	2005	0	98133	47.7274	-122.357	
...	
21586	0	2009	0	98107	47.6707	-122.381	
21588	190	2007	0	98117	47.6756	-122.375	
21595	50	2006	0	98103	47.6533	-122.346	
21610	0	2009	0	98144	47.5944	-122.299	
21612	0	2008	0	98144	47.5941	-122.299	

	sqft_living15	sqft_lot15
2	2720	8062
11	1330	6000
18	1060	5095

23	1220	8386
31	1390	1756
...
21586	1430	1249
21588	1210	1118
21595	1530	1282
21610	1020	2007
21612	1020	1357

[2972 rows x 21 columns]

```
[ ]: print(f"Count: {selected_houses.shape[0]}")
```

Count: 2972

11. Display houses having number of bedrooms more than 2 and square foot

```
[ ]: selected_houses = df[df['bedrooms'] > 2][['bedrooms', 'sqft_living']]
```

```
[ ]: selected_houses
```

```
[ ]:
      bedrooms  sqft_living
0           3         1180
1           3         2570
3           4         1960
4           3         1680
5           4         5420
...
21606        4         3510
21607        3         1310
21608        3         1530
21609        4         2310
21611        3         1600
```

[18641 rows x 2 columns]

12. Replace/eliminate missing values

```
[ ]: df = df.dropna()
```

```
[ ]: df.shape[0]
```

```
[ ]: 21613
```

13. Change column name(s) to short/easy names if required.

```
[ ]: df.columns
```

```
[ ]: Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
          'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
          'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
          'lat', 'long', 'sqft_living15', 'sqft_lot15'],
          dtype='object')
```

```
[ ]: df = df.rename(columns={
          'sqft_living': 'living_area'
        })
```

```
[ ]: df.columns
```

```
[ ]: Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'living_area',
          'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
          'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
          'lat', 'long', 'sqft_living15', 'sqft_lot15'],
          dtype='object')
```

```
[ ]: df
```

```
[ ]:
      id      date  price  bedrooms  bathrooms  living_area \
0  7129300520  20141013T000000  221900         3         1.00      1180
1  6414100192  20141209T000000  538000         3         2.25      2570
2  5631500400  20150225T000000  180000         2         1.00       770
3  2487200875  20141209T000000  604000         4         3.00      1960
4  1954400510  20150218T000000  510000         3         2.00      1680
...
21608  263000018  20140521T000000  360000         3         2.50      1530
21609  6600060120  20150223T000000  400000         4         2.50      2310
21610  1523300141  20140623T000000  402101         2         0.75      1020
21611   291310100  20150116T000000  400000         3         2.50      1600
21612  1523300157  20141015T000000  325000         2         0.75      1020

      sqft_lot  floors  waterfront  view  ...  grade  sqft_above  \
0         5650     1.0           0     0  ...     7         1180
1         7242     2.0           0     0  ...     7         2170
2        10000     1.0           0     0  ...     6          770
3         5000     1.0           0     0  ...     7        1050
4         8080     1.0           0     0  ...     8        1680
...
21608        1131     3.0           0     0  ...     8        1530
21609         5813     2.0           0     0  ...     8        2310
21610         1350     2.0           0     0  ...     7        1020
21611         2388     2.0           0     0  ...     8        1600
21612         1076     2.0           0     0  ...     7        1020

      sqft_basement  yr_built  yr_renovated  zipcode      lat      long  \
```

0	0	1955	0	98178	47.5112	-122.257
1	400	1951	1991	98125	47.7210	-122.319
2	0	1933	0	98028	47.7379	-122.233
3	910	1965	0	98136	47.5208	-122.393
4	0	1987	0	98074	47.6168	-122.045
...
21608	0	2009	0	98103	47.6993	-122.346
21609	0	2014	0	98146	47.5107	-122.362
21610	0	2009	0	98144	47.5944	-122.299
21611	0	2004	0	98027	47.5345	-122.069
21612	0	2008	0	98144	47.5941	-122.299

	sqft_living15	sqft_lot15
0	1340	5650
1	1690	7639
2	2720	8062
3	1360	5000
4	1800	7503
...
21608	1530	1509
21609	1830	7200
21610	1020	2007
21611	1410	1287
21612	1020	1357

[21613 rows x 21 columns]

14. Drop unessential columns (feature selection).

```
[ ]: df = df.drop(['id', 'waterfront', 'view'], axis=1)
```

```
[ ]: df.columns
```

```
[ ]: Index(['date', 'price', 'bedrooms', 'bathrooms', 'living_area', 'sqft_lot',
          'floors', 'condition', 'grade', 'sqft_above', 'sqft_basement',
          'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15',
          'sqft_lot15'],
          dtype='object')
```

15. Find mean/min/max of numeric columns.

```
[ ]: df.mean()
```

/home/oneautumleaf/tmp/ipykernel_6892/3698961737.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.mean()
```

```
[ ]: price          540088.141905
     bedrooms        3.370842
     bathrooms        2.114757
     living_area      2079.899736
     sqft_lot         15106.967566
     floors           1.494309
     condition        3.409430
     grade            7.656873
     sqft_above       1788.390691
     sqft_basement     291.509045
     yr_built         1971.005136
     yr_renovated      84.402258
     zipcode          98077.939805
     lat              47.560053
     long             -122.213896
     sqft_living15     1986.552492
     sqft_lot15        12768.455652
     dtype: float64
```

```
[ ]: df.min()
```

```
[ ]: date          20140502T000000
     price          75000
     bedrooms        0
     bathrooms        0.0
     living_area      290
     sqft_lot         520
     floors           1.0
     condition        1
     grade            1
     sqft_above       290
     sqft_basement     0
     yr_built         1900
     yr_renovated      0
     zipcode          98001
     lat              47.1559
     long             -122.519
     sqft_living15     399
     sqft_lot15        651
     dtype: object
```

```
[ ]: df.max()
```

```
[ ]: date          20150527T000000
     price          7700000
```

```

bedrooms          33
bathrooms         8.0
living_area       13540
sqft_lot          1651359
floors            3.5
condition         5
grade            13
sqft_above        9410
sqft_basement     4820
yr_built          2015
yr_renovated      2015
zipcode           98199
lat               47.7776
long              -121.315
sqft_living15     6210
sqft_lot15        871200
dtype: object

```

16. Find mode of all columns.

```
[ ]: df.mode()
```

```

[ ]:
   date      price  bedrooms  bathrooms  living_area  sqft_lot  \
0  20140623T000000  350000.0      3.0        2.5      1300.0    5000.0
1              NaN  450000.0      NaN        NaN         NaN         NaN
2              NaN      NaN      NaN        NaN         NaN         NaN
3              NaN      NaN      NaN        NaN         NaN         NaN

   floors  condition  grade  sqft_above  sqft_basement  yr_built  \
0      1.0         3.0    7.0      1300.0           0.0    2014.0
1      NaN         NaN    NaN         NaN           NaN         NaN
2      NaN         NaN    NaN         NaN           NaN         NaN
3      NaN         NaN    NaN         NaN           NaN         NaN

   yr_renovated  zipcode    lat    long  sqft_living15  sqft_lot15
0           0.0  98103.0  47.5322 -122.29      1540.0      5000.0
1           NaN     NaN  47.5491    NaN         NaN         NaN
2           NaN     NaN  47.6624    NaN         NaN         NaN
3           NaN     NaN  47.6846    NaN         NaN         NaN

```

17. Display unique values in each column

```
[ ]: for column in df.columns:
      print(f"{column}: {len(df[column].unique())}")
```

```

date: 372
price: 4032
bedrooms: 13

```

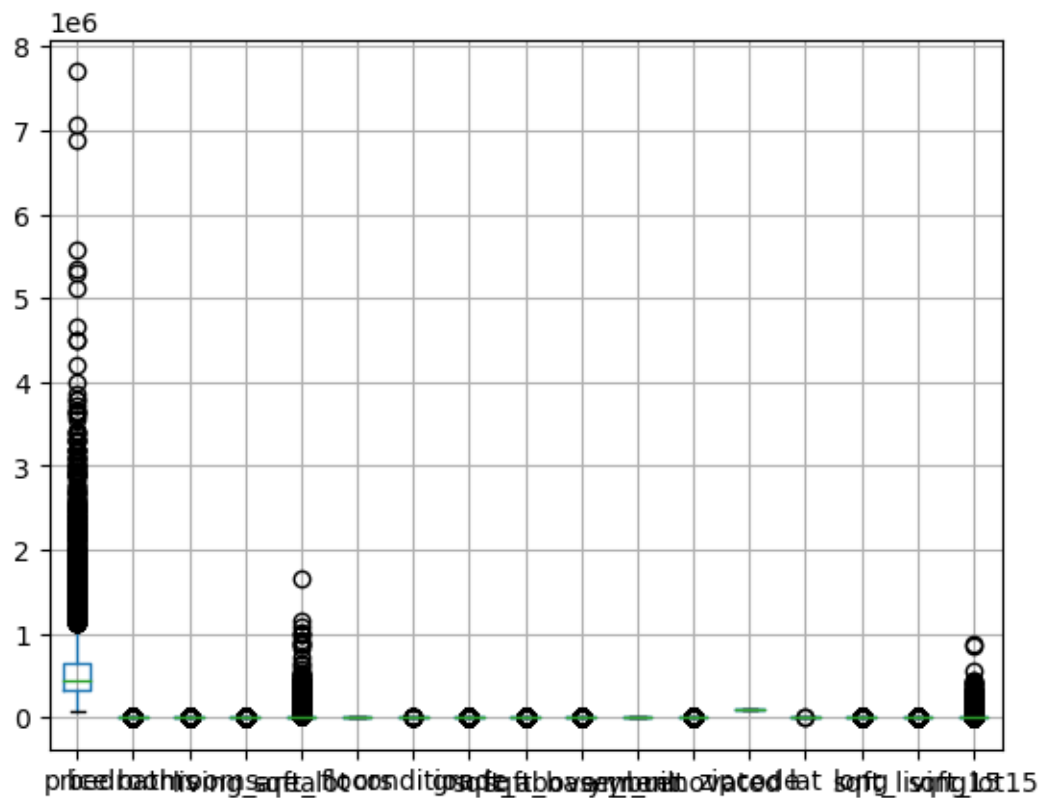


```

bathrooms: 30
living_area: 1038
sqft_lot: 9782
floors: 6
condition: 5
grade: 12
sqft_above: 946
sqft_basement: 306
yr_built: 116
yr_renovated: 70
zipcode: 70
lat: 5034
long: 752
sqft_living15: 777
sqft_lot15: 8689

```

```
[ ]: df.boxplot()
plt.show()
```



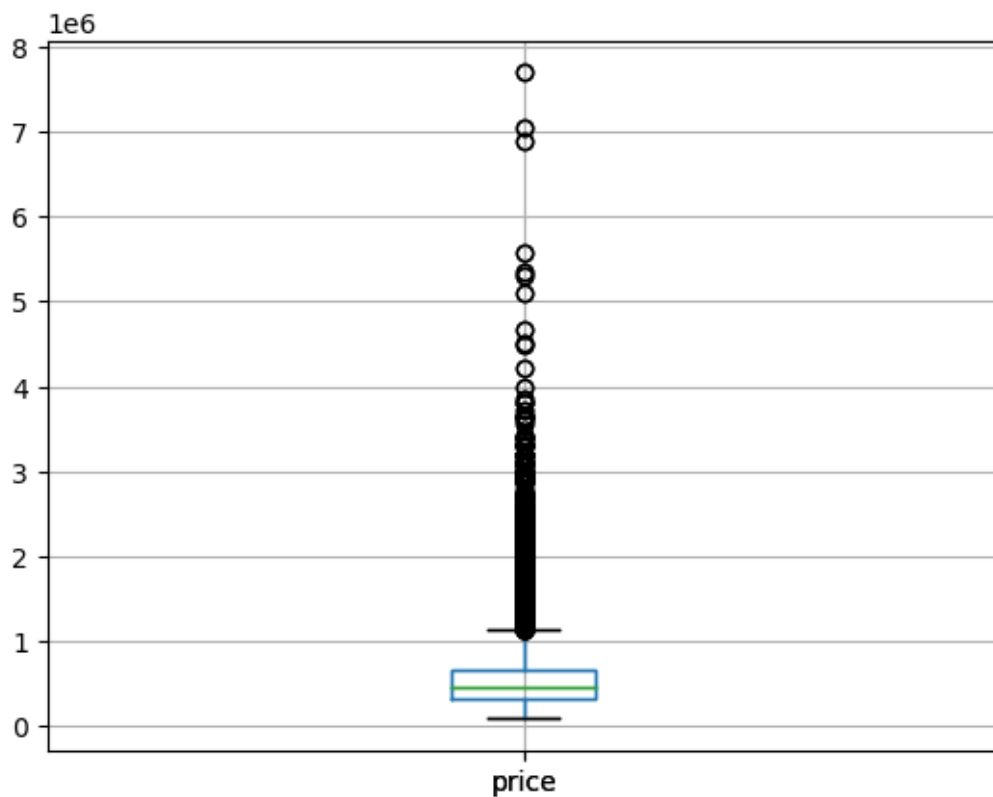
```
[ ]: df.quantile([0, .25, .5, .75, 1], numeric_only=True)
```

```
[ ]:
      price  bedrooms  bathrooms  living_area  sqft_lot  floors  \
0.00   75000.0        0.0        0.00       290.0    520.0     1.0
0.25  321950.0        3.0        1.75      1427.0   5040.0     1.0
0.50  450000.0        3.0        2.25      1910.0   7618.0     1.5
0.75  645000.0        4.0        2.50      2550.0  10688.0     2.0
1.00 7700000.0       33.0        8.00     13540.0 1651359.0     3.5

      condition  grade  sqft_above  sqft_basement  yr_built  yr_renovated  \
0.00         1.0    1.0       290.0           0.0    1900.0           0.0
0.25         3.0    7.0      1190.0           0.0    1951.0           0.0
0.50         3.0    7.0      1560.0           0.0    1975.0           0.0
0.75         4.0    8.0      2210.0          560.0    1997.0           0.0
1.00         5.0   13.0      9410.0         4820.0    2015.0        2015.0

      zipcode      lat      long  sqft_living15  sqft_lot15
0.00  98001.0  47.1559 -122.519         399.0      651.0
0.25  98033.0  47.4710 -122.328        1490.0     5100.0
0.50  98065.0  47.5718 -122.230        1840.0     7620.0
0.75  98118.0  47.6780 -122.125        2360.0    10083.0
1.00  98199.0  47.7776 -121.315        6210.0    871200.0
```

```
[ ]: df.boxplot('price')
plt.show()
```



```
[ ]: df.quantile([.5])['price']
```

```
/home/oneautumleaf/tmp/ipykernel_6892/1945159076.py:1: FutureWarning: The
default value of numeric_only in DataFrame.quantile is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
```

```
df.quantile([.5])['price']
```

```
[ ]: 0.5    450000.0
      Name: price, dtype: float64
```

```
[ ]:
```