Problem statement:

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, you want to sell a house and you don't know the price which you may expect — it can't be too low or too high. To find house price you usually try to find similar properties in your neighborhood and based on gathered data you will try to assess your house price.

Objective:

Take advantage of all of the feature variables available below, use it to analyse and predict house prices.

1. cid: a notation for a house
2. dayhours: Date house was sold
3. price: Price is prediction target
4. room_bed: Number of Bedrooms/House
5. room_bath: Number of bathrooms/bedrooms
6. living_measure: square footage of the home
7. lot_measure: quare footage of the lot
8. ceil: Total floors (levels) in house
9. coast: House which has a view to a waterfront
10. sight: Has been viewed
11. condition: How good the condition is (Overall)
12. quality: grade given to the housing unit, based on grading system
13. ceil_measure: square footage of house apart from basement
14. basement_measure: square footage of the basement
15. yr_built: Built Year
16. yr_renovated: Year when house was renovated
17. zipcode: zip
18. lat: Latitude coordinate
19. long: Longitude coordinate
20. living_measure15: Living room area in 2015(implies-- some renovations) This might or might not have affected the lotsize area
21. lot_measure15: lotSize area in 2015(implies-- some renovations)
22. furnished: Based on the quality of room
23. total_area: Measure of both living and lot

Table :

	cid	dayhours	price	room_bed	room_bath	living_measure	lot_measure	ceil	coast	sight	condition	quality	ceil_measure	basement
	3.876101e+09	20150427T000000	600000.0	4.0	1.75	3050.0	9440.0	1.0	0.0	0.0	3.0	8.0	1800.0	1250.0
	3.145600e+09	20150317T000000	190000.0	2.0	1.00	670.0	3101.0	1.0	0.0	0.0	4.0	6.0	670.0	0.0
	7.129303e+09	20140820T000000	735000.0	4.0	2.75	3040.0	2415.0	2.0	1.0	4.0	3.0	8.0	3040.0	0.0
	7.338220e+09	20141010T000000	257000.0	3.0	2.50	1740.0	3721.0	2.0	0.0	0.0	3.0	8.0	1740.0	0.0
	7.950301e+09	20150218T000000	450000.0	2.0	1.00	1120.0	4590.0	1.0	0.0	0.0	3.0	7.0	1120.0	0.0

yr_built	yr_renovated	zipcode	lat	long	living_measure15	lot_measure15	furnished	total_area
1966.0	0.0	98034.0	47.7228	-122.183	2020.0	8660.0	0.0	12490.0
1948.0	0.0	98118.0	47.5546	-122.274	1660.0	4100.0	0.0	3771.0
1966.0	0.0	98118.0	47.5188	-122.256	2620.0	2433.0	0.0	5455.0
2009.0	0.0	98002.0	47.3363	-122.213	2030.0	3794.0	0.0	5461.0
1924.0	0.0	98118.0	47.5663	-122.285	1120.0	5100.0	0.0	5710.0

SHAPE:

```
(21613, 23)
```

INFO:

```
#    Column             Non-Null  Count   Dtype
---  ------              --------------   -----
 0   dayhours            21613 non-null   object
 1   price               21613 non-null   float64
 2   room_bed            21505 non-null   float64
 3   room_bath           21505 non-null   float64
 4   living_measure      21596 non-null   float64
 5   lot_measure         21571 non-null   float64
 6   ceil                21571 non-null   object
 7   coast               21612 non-null   object
 8   sight               21556 non-null   float64
 9   condition           21556 non-null   object
 10  quality             21612 non-null   float64
 11  ceil_measure        21612 non-null   float64
 12  basement            21612 non-null   float64
 13  yr_built            21612 non-null   object
 14  yr_renovated        21613 non-null   float64
 15  zipcode             21613 non-null   float64
 16  lat                 21613 non-null   float64
 17  long                21613 non-null   object
 18  living_measure15    21447 non-null   float64
 19  lot_measure15       21584 non-null   float64
 20  furnished           21584 non-null   float64
 21  total_area          21584 non-null   object
```

CHECKING NULL VALUES :

```
dayhours             0
price                0
room_bed           108
room_bath          108
living_measure      17
lot_measure         42
ceil                42
coast                1
sight               57
condition           57
quality              1
ceil_measure         1
basement             1
yr_built             1
yr_renovated         0
zipcode              0
lat                  0
long                 0
living_measure15   166
lot_measure15       29
furnished           29
total_area          29
```

VALUE COUNTS :

We have atleast 13 variables containing NaN & $ Values in them, they are as follows :

1.room_bed

```
([ 4.,  2.,  3.,  1.,  5.,  6., nan,  7., 10.,  8.,  0.,  9., 33.,11.])
```

2.room_bath

```
([1.75, 1.  , 2.75, 2.5 , 1.5 , 3.5 , 2.  , 2.25, 3.  , 4.  , 3.25,3.75,  nan, 5.
, 0.75, 5.5 , 4.25, 4.5 , 4.75, 8.  , 6.75, 5.25,
       6.  , 0.  , 1.25, 5.75, 7.5 , 6.5 , 0.5 , 7.75, 6.25])

3. living_measure
([3050.,  670., 3040., ..., 1405., 1295., 2253.])

4.lot_measure
([ 9440.,  3101.,  2415., ..., 12369.,  2332., 60467.])

5. sight
([ 0.,  4.,  2.,  3.,  1., nan])

6. quality
([ 8.,  6.,  7., 10.,  9.,  5., 11., 13.,  4., 12.,  1.,  3., nan])

7.basement
([1250.,    0., 1320., 1000.,  480.,  610., 1050.,  700.,  430.,
        560.,  250.,  670.,  570.,  290.,  600.,  680.,  380.,   50.,
       1020.,  690., 1010.,  530., 1370., 1040.,  790.,  910.,  820.,
       1850.,  500.,  760.,  960.,  340.,  800.,  580., 1600., 1680.,
        900.,  420.,  450.,  200.,  240.,  950., 1590., 1220., 1500.,
```

```
        710.,   80.,  140., 1260.,  860.,  890.,  280.,  440.,  880.,
        220., 1650.,  630.,  780.,  810.,  300.,  720.,  470.,  150.,
       1180., 1060.,  120.,  660.,  400., 1100., 1780.,  640., 1170.,
       1890.,  130.,  550.,  360.,  940.,  650., 2730.,  870.,  730.,
       1350., 1530., 1540.,  620., 1080., 1900.,  770.,  520.,  920.,
       1110.,  830., 1420.,  980.,  190.,  330.,  350.,  740., 1570.,
        990., 1390.,  260.,  540., 1300.,  265., 1120.,  460.,  370.,
       1830., 1140.,  270.,  145.,  510.,  750., 1710.,  930., 1870.,
       1200.,  310.,  850.,  506.,  970., 1070., 1450.,  840.,   90.,
       3500., 1380., 1090., 1280., 1240., 3480., 1210., 1690.,   60.,
       1800., 2400.,  180., 4820.,  110., 1030., 2060.,  143., 1400.,
        100., 1270., 2040., 1360., 1740.,  590., 1150.,   40., 1990.,
       1340., 1700.,  160., 1290., 1190., 1630.,  946., 1230., 1430.,
       2600.,  390., 1620.,  410., 1950., 1160., 1135.,  320.,  210.,
       1460.,  170., 1490., 1330., 1760.,  207., 2300., 1410., 2090.,
       1810., 1660., 1940., 3260., 1640.,  894., 1440., 2200., 1130.,
       2010., 1790.,  490., 1550., 1560.,  230.,   70.,  276.,  417.,
        652., 2000.,  283., 1580., 1670., 1310., 1720., 2390., 2100.,
        374.,  414., 2620.,  176., 1910.,  515., 1730., 1820., 2080.,
        666., 1480.,  861., 1520., 1470., 1816.,  518.,  784.,   10.,
       2110., 2050., 4130., 1008., 2330., 2030.,  516.,  704., 2580.,
        915.,  172., 1510.,  602., 2550., 1610., 1284., 1281., 2170.,
       1798., 2240., 2070., 1930., 1880., 2020.,  508.,  295., 2360.,
       2720., 2160.,  435.,  225., 2220., 1860., 1840., 2590., 2130.,
       2490.,  862., 3000., 2310., 2150.,  556., 1852.,  475., 1548.,
       1960.,  235., 2610.,  875., 1024., 2190.,  415.,  792.,  768.,
       1248., 1275.,   20., 2850., 1525., 2120., 1913., 2250.,   65.,
       1770., 1750., 2570., 2500.,  588.,  266., 2350., 1481.,  274.,
        248.,  935., 1245., 2196.,  243., 2810.,   nan,  906., 1920.,
       2180.])
```

8.furnished

```
([ 0.,  1., nan])
```

9.ceil

```
([1.0, 2.0, 3.0, 1.5, 2.5, '$', nan, 3.5])
```

10. coast

```
[0.0, 1.0, '$', nan]
```

11. condition

```
[3.0, 4.0, 5.0, 2.0, nan, 1.0, '$']
```

12.yr_built

```
([1966.0, 1948.0, 2009.0, 1924.0, 1994.0, 2005.0, 1978.0, 1983.0,
       2012.0, 1912.0, 1990.0, 1967.0, 1919.0, 1908.0, 1950.0, 2000.0,
       2013.0, 1943.0, 1922.0, 1977.0, 2004.0, 1935.0, 1964.0, 1945.0,
       1987.0, 2008.0, 1940.0, 2003.0, 1988.0, 1985.0, 1998.0, 1995.0,
       1946.0, 1984.0, 1958.0, 1963.0, 1942.0, 2014.0, 1971.0, 1936.0,
       1954.0, 1923.0, 2002.0, 1972.0, 2007.0, 1930.0, 1962.0, 1999.0,
       1953.0, 1965.0, 2010.0, 1997.0, 2006.0, 1979.0, 1996.0, 1992.0,
       1968.0, 1980.0, 1981.0, 1969.0, 2001.0, 1929.0, 1952.0, 1916.0,
       1976.0, 1974.0, 1920.0, 1931.0, 1975.0, 1960.0, 1900.0, '$',
       1986.0, 1989.0, 1906.0, 1955.0, 1956.0, 1915.0, 1941.0, 1993.0,
       2011.0, 1925.0, 1947.0, 1991.0, 1926.0, 1927.0, 1951.0, 1961.0,
       1932.0, 1917.0, 1928.0, 1959.0, 1921.0, 1911.0, 1949.0, 1982.0,
       1913.0, 1957.0, 1914.0, 1938.0, 1973.0, 1937.0, 1944.0, 1970.0,
       1901.0, 1907.0, 1939.0, 1918.0, 1934.0, 1904.0, 2015.0, 1909.0,
       1910.0, 1905.0, 1902.0, 1933.0, 1903.0, nan])
```

13. total_area

```
([12490.0, 3771.0, 5455.0, ..., 16111.0, 63597.0, 38122.0])
```

As we can see here there are $ & NaN values present in our data we need to impute them.

Replacing $ with NaN from these value so that we can get null values and can further remove them by applying KNNImputer Method :

["ceil","coast","condition","yr_built","total_area","long"]

Now, all the $ sign have been removed with NaN Values, but we know there are still NaN values present in our data which we are going to impute through KNNImputer.

Removing 'T000000' from dayhour variable so that we can get the dates

Now we have removed the values with the help of KNNImputer

INFO:

```
#   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   dayhours           21613 non-null  float64
 1   price              21613 non-null  float64
 2   room_bed           21613 non-null  float64
 3   room_bath          21613 non-null  float64
 4   living_measure     21613 non-null  float64
 5   lot_measure        21613 non-null  float64
 6   ceil               21613 non-null  float64
 7   coast              21613 non-null  float64
 8   sight              21613 non-null  float64
 9   condition          21613 non-null  float64
 10  quality            21613 non-null  float64
 11  ceil_measure       21613 non-null  float64
 12  basement           21613 non-null  float64
 13  yr_built           21613 non-null  float64
 14  yr_renovated       21613 non-null  float64
 15  living_measure15   21613 non-null  float64
 16  lot_measure15      21613 non-null  float64
 17  furnished          21613 non-null  float64
 18  total_area         21613 non-null  float64
 19  lat                21613 non-null  float64
 20  long               21613 non-null  float64
 21  zipcode            21613 non-null  float64
```

Here, we can see all of the values have been imputed with help of KNNIm puter. I've used specifically this imputer because there was chance that o ur model could get biased if we had used MEAN,MEDIAN,MODE formula but in case of this imputer it searches for the nearest value and then impu te those values with it.

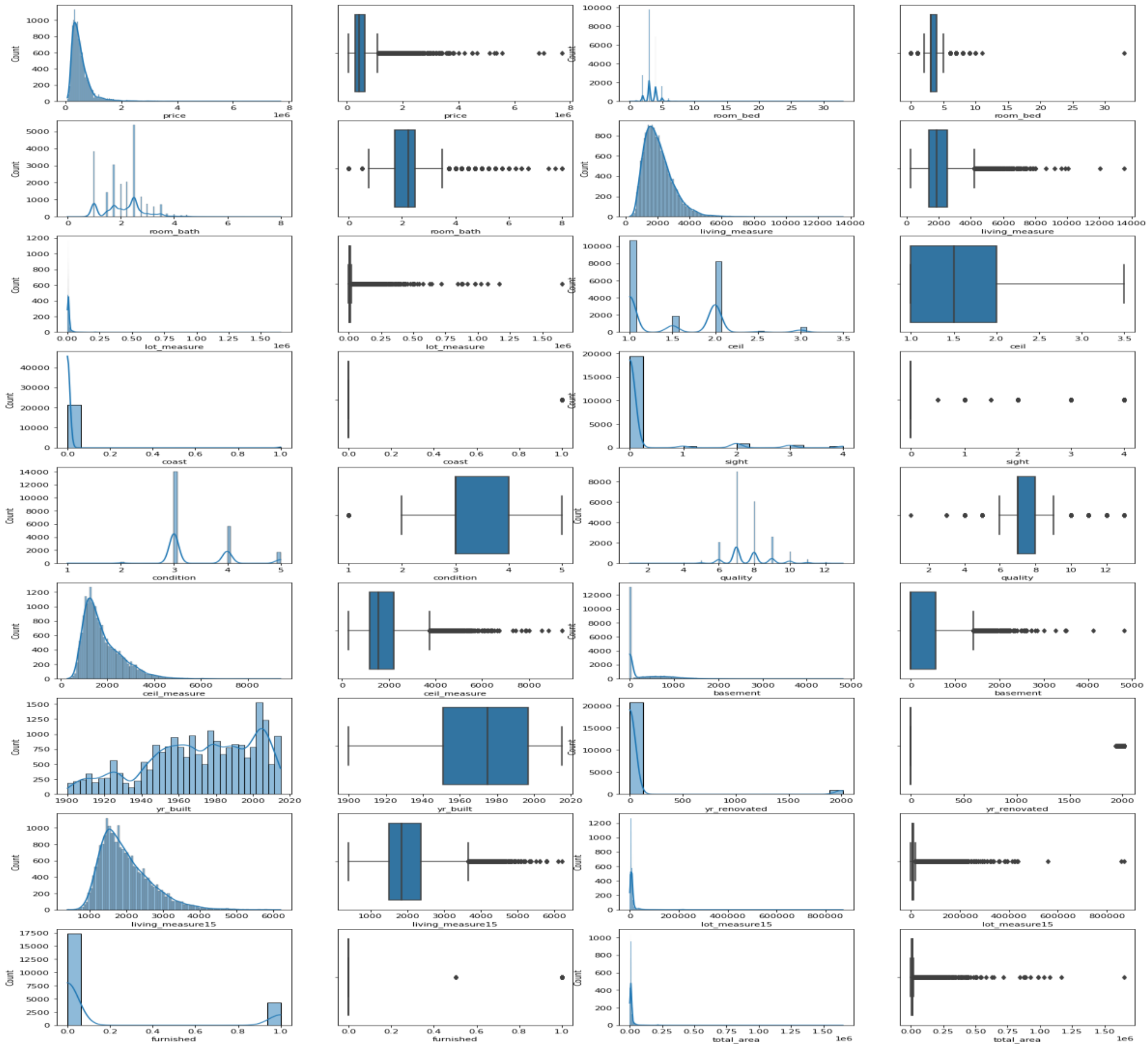
Description :

	count	mean	std	min	25%	50%	75%	max
dayhours	21613.0	2.014390e+07	4436.582469	2.014050e+07	2.014072e+07	2.014102e+07	2.015022e+07	2.015053e+07
price	21613.0	5.401822e+05	367362.231718	7.500000e+04	3.219500e+05	4.500000e+05	6.450000e+05	7.700000e+06
room_bed	21613.0	3.371582e+00	0.929343	0.000000e+00	3.000000e+00	3.000000e+00	4.000000e+00	3.300000e+01
room_bath	21613.0	2.115168e+00	0.769351	0.000000e+00	1.750000e+00	2.250000e+00	2.500000e+00	8.000000e+00
living_measure	21613.0	2.079903e+03	918.300749	2.900000e+02	1.430000e+03	1.910000e+03	2.550000e+03	1.354000e+04
lot_measure	21613.0	1.509801e+04	41389.711890	5.200000e+02	5.040000e+03	7.620000e+03	1.068800e+04	1.651359e+06
ceil	21613.0	1.494182e+00	0.539604	1.000000e+00	1.000000e+00	1.500000e+00	2.000000e+00	3.500000e+00
coast	21613.0	7.449220e-03	0.085989	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
sight	21613.0	2.344885e-01	0.765929	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.000000e+00
condition	21613.0	3.409244e+00	0.650148	1.000000e+00	3.000000e+00	3.000000e+00	4.000000e+00	5.000000e+00
quality	21613.0	7.656873e+00	1.175459	1.000000e+00	7.000000e+00	7.000000e+00	8.000000e+00	1.300000e+01
ceil_measure	21613.0	1.788347e+03	828.088623	2.900000e+02	1.190000e+03	1.560000e+03	2.210000e+03	9.410000e+03
basement	21613.0	2.915343e+02	442.573959	0.000000e+00	0.000000e+00	0.000000e+00	5.600000e+02	4.820000e+03
yr_built	21613.0	1.971007e+03	29.366925	1.900000e+03	1.951000e+03	1.975000e+03	1.997000e+03	2.015000e+03
yr_renovated	21613.0	8.440226e+01	401.679240	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.015000e+03
living_measure15	21613.0	1.986686e+03	684.476238	3.990000e+02	1.490000e+03	1.840000e+03	2.360000e+03	6.210000e+03
lot_measure15	21613.0	1.277500e+04	27310.371557	6.510000e+02	5.100000e+03	7.620000e+03	1.008700e+04	8.712000e+05
furnished	21613.0	1.966178e-01	0.397406	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
total_area	21613.0	1.718856e+04	41595.794198	1.423000e+03	7.032000e+03	9.575000e+03	1.300000e+04	1.652659e+06
lat	21613.0	4.756005e+01	0.138564	4.715590e+01	4.747100e+01	4.757180e+01	4.767800e+01	4.777760e+01
long	21613.0	-1.222139e+02	0.140851	-1.225190e+02	-1.223280e+02	-1.222310e+02	-1.221250e+02	-1.213150e+02
zipcode	21613.0	9.807794e+04	53.505026	9.800100e+04	9.803300e+04	9.806500e+04	9.811800e+04	9.819900e+04

1. **CID:** House ID/Property ID.Not used for analysis
2. **Dayhours:** 5 factor analysis is reflecting for this column
3. **price:** Our taget column value is in 75k - 7700k range. As Mean > Median, it's **rightly skewed**.
4. **room_bed:** Number of bedrooms range from 0 - 33. As Mean slightly > Median, it's **slightly rightly skewed.**
5. **room_bath:** Number of bathrooms range from 0 - 8. As Mean slightly < Median, it's **slightly leftly skewed**.
6. **living_measure:** Square footage of house range from 290 - 13,540. As Mean > Median, it's **rightly skewed**.
7. **lot_measure:** Square footage of lot range from 520 - 16,51,359. As Mean almost double of Median, it's **Hightly rightly skewed**.
8. **ceil:** Number of floors range from 1 - 3.5 As Mean ~ Median, it's **almost Normal Distributed**.
9. **coast:** As this value represent whether house has waterfront view or not. It's **categorical column**. From above analysis we got know, very few houses has waterfront view.
10. **sight:** Value ranges from 0 - 4. As Mean > Median, it's **rightly skewed**
11. **condition:** Represents rating of house which ranges from 1 - 5. As Mean > Median, it's **rightly skewed**
12. **quality:** Representign grade given to house which range from 1 - 13. As Mean > Median, it's **rightly skewed**.

13. **ceil_measure:** Square footage of house apart from basement ranges in 290 - 9,410. As Mean > Median, it's **rightly skewed**.
14. **basement:** Square footage house basement ranges in 0 - 4,820. As Mean highlty > Median, it's **Highly rightly skewed**.
15. **yr_built:** House built year ranges from 1900 - 2015. As Mean < Median, it's **leftly skewed**.
16. **yr_renovated:** House renovation year only 2015. So this column can be used as **Categorical Variable** for knowing whether house is renovated or not.
17. **zipcode:** House ZipCode ranges from 98001 - 98199. As Mean > Median, it's **rightly skewed**.
18. **lat:** Lattitude ranges from 47.1559 - 47.7776 As Mean < Median, it's **leftly skewed**.
19. **long:** Longittude ranges from -122.5190 to -121.315 As Mean > Median, it's **rightly skewed**.
20. **living_measure15:** Value ragnes from 399 to 6,210. As Mean > Median, it's **rightly skewed**.
21. **lot_measure15:** Value ragnes from 651 to 8,71,200. As Mean highly > Median, it's **Highly rightly skewed**.
22. **furnished:** Representing whether house is furnished or not. It's a **Categorical Variable**
23. **total_area** Total area of house ranges from 1,423 to 16,52,659. As Mean is almost double of Median, it's **Highly rightly skewed**

Checking for Outliers:



Through the help of Boxplot & Histogram we can see which variables have outliers in them and then remove those outliers with the help of IQR method. So, here we can see that 6 variables have outliers in them

1. room_bed.

2. living_measure.

3. lot_measure.

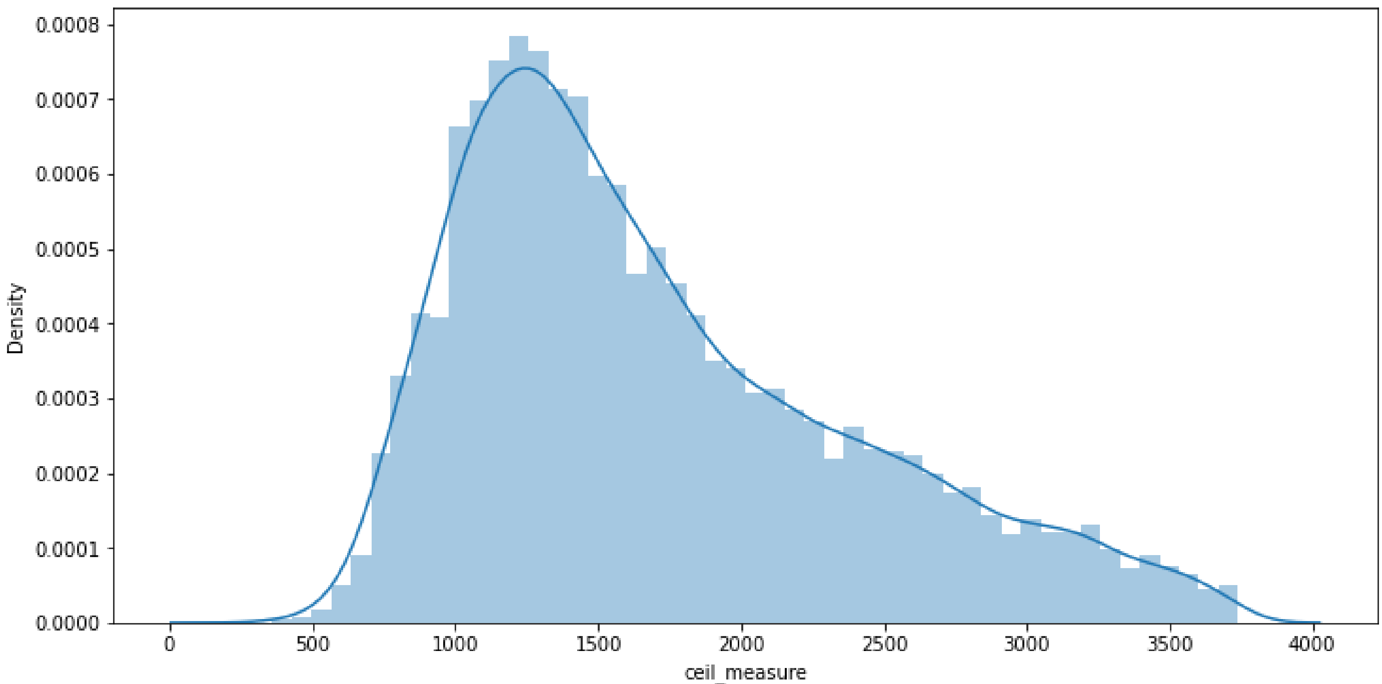
4. ceil_measure.

5. basement.

Removing Outliers throught IQR method

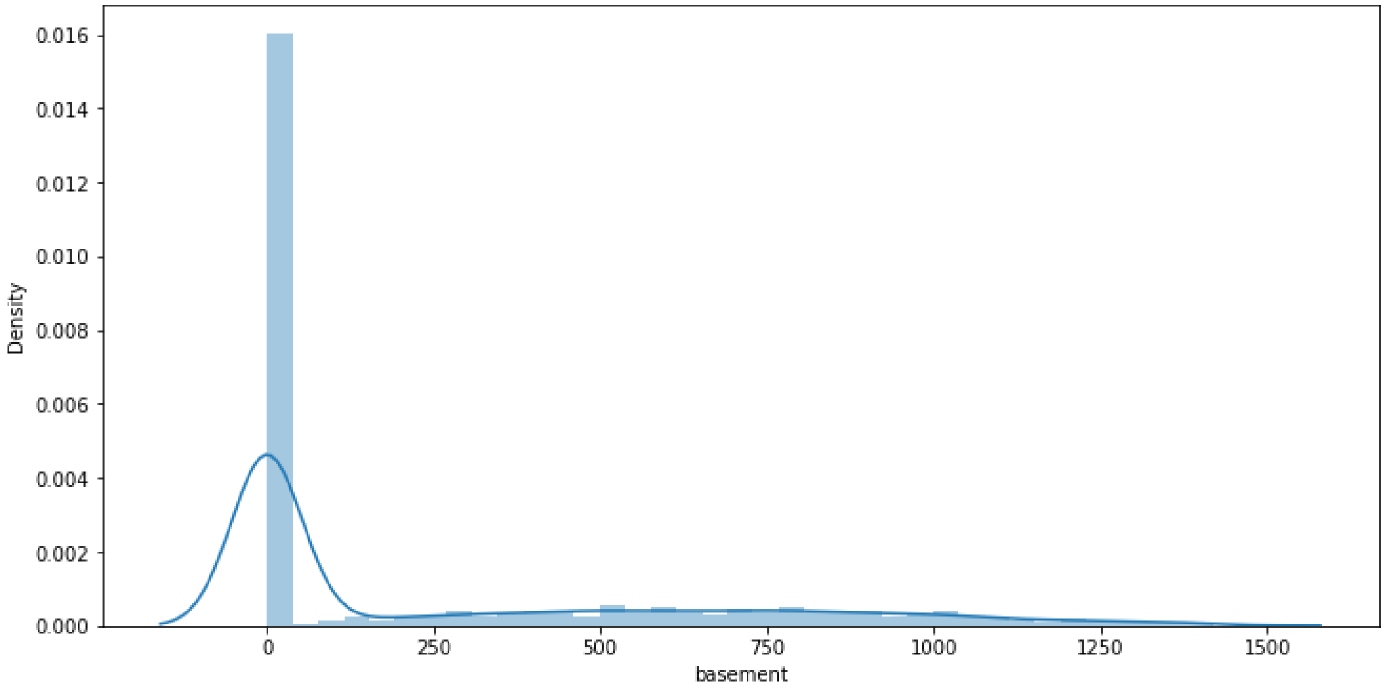
CEIL_MEASURE :

We got 611 records which are outliers from ceil_measure variable which have been removed and now the shape of the data has been reduced to 21002 rows & 22 Columns, After treating outliers o f ceil_measure, the data has reduced by about 600(~3%) data points but data is nicely distributed.



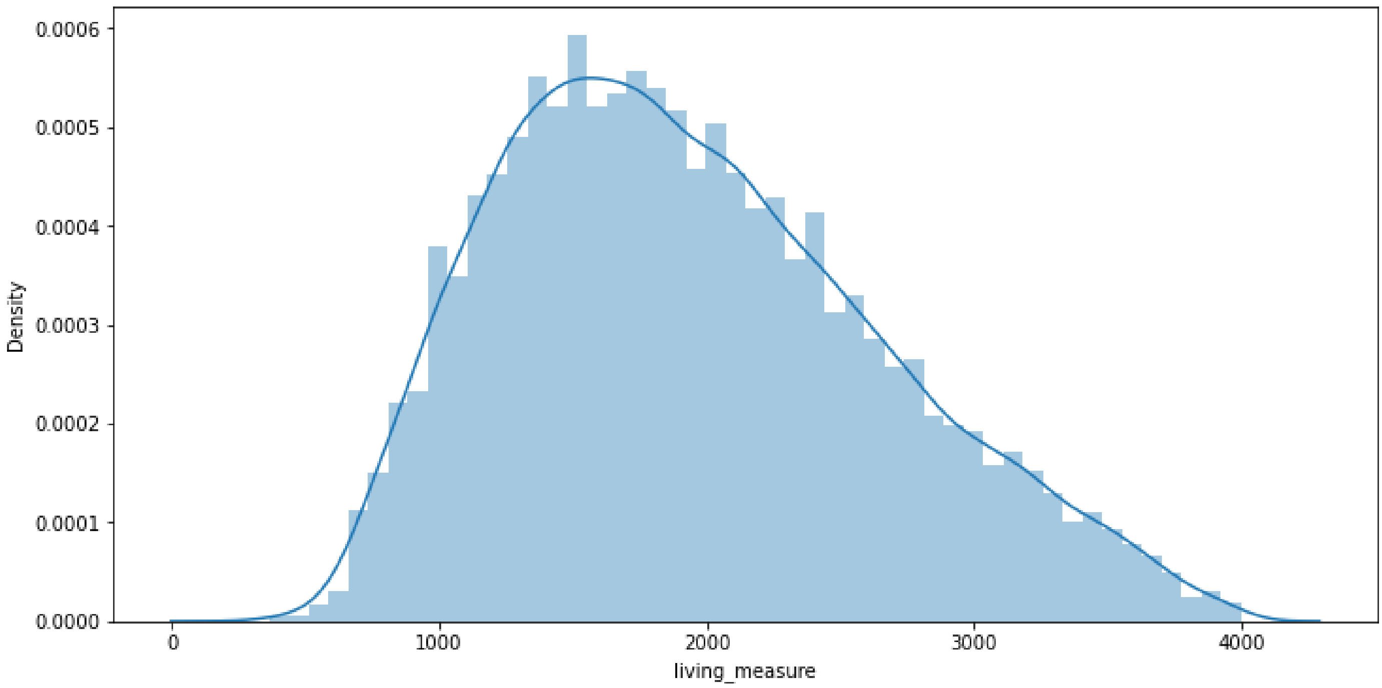
BASEMENT :

We get 408 records which are outliers from basement variable which have been removed and now the shape of the data has been reduced to 20594 rows & 22 columns, After treating outliers of basement, we can see that 400(~2%) data points got imputed. Total about 5% data has been imputed after treating ceil_measure and basement.



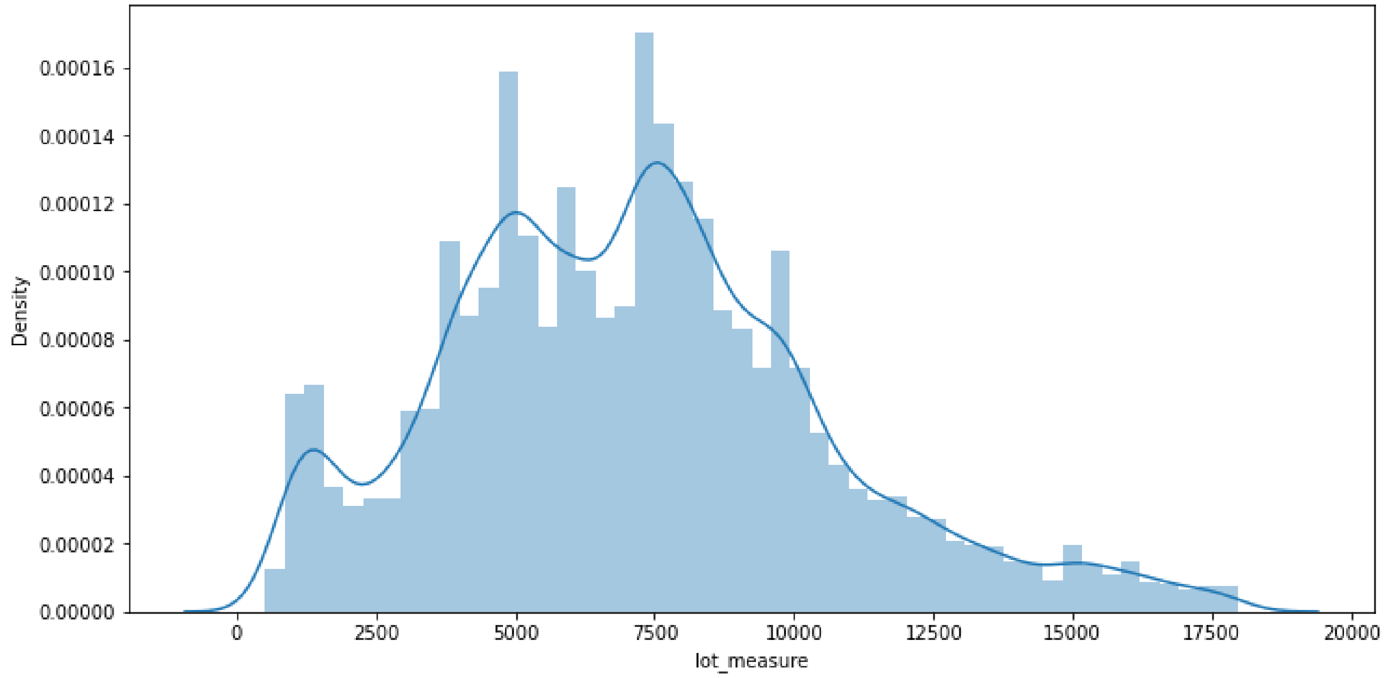
living_measure

We get 178 records which are outliers from living_measure variable which have been removed and now the shape of the data has been reduced to 20416 rows & 22 columns, After treating outliers of living_measure, we deducted 178 data points more and data distribution looks normal.



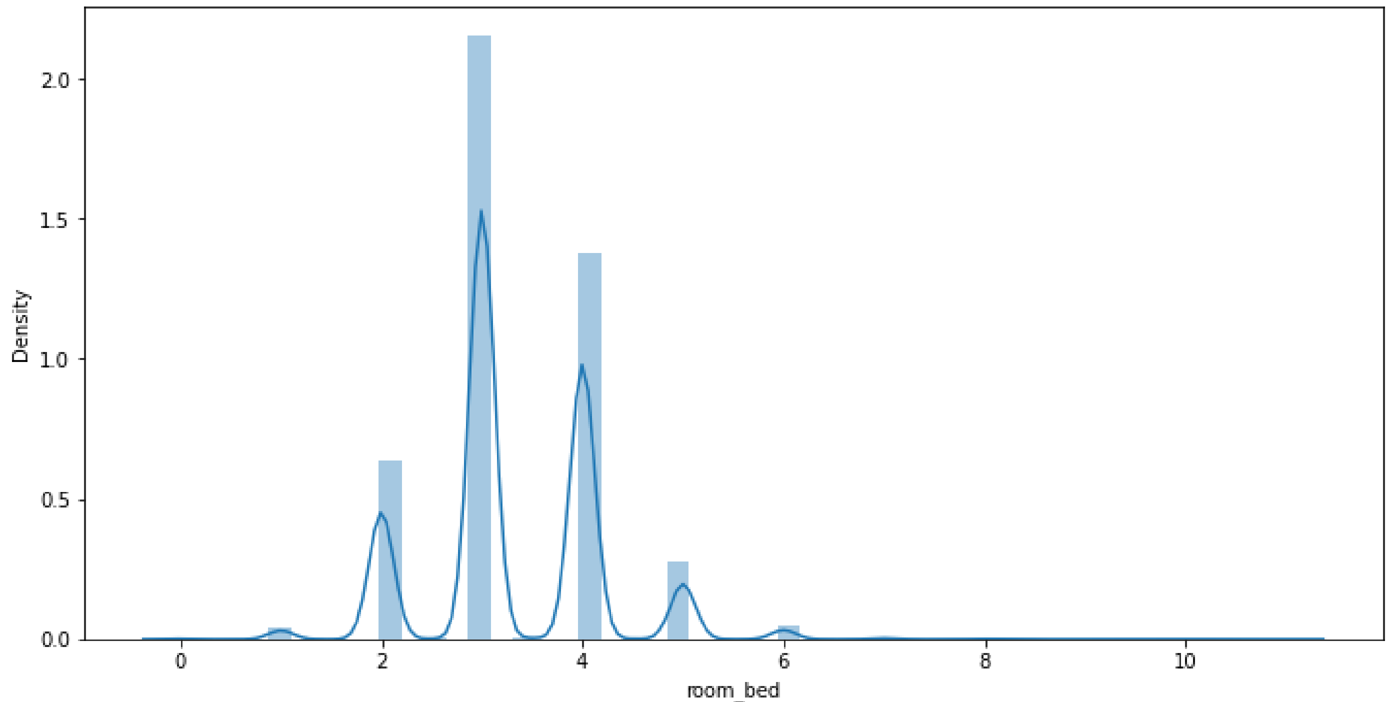
lot_measure

We get 2126 records which are outliers from lot_measure variable which have been removed and now the shape of the data has been reduced to 18290 rows & 22 columns, Total outliers in the lot_measure are 2126 data points. But still we are going ahead with imputing the data. We will analyze later whether there is any impact on the data set or not.



room_bed

For room_bed variable there is only one outlier which needs to be treated and after treating the outliers the data which we have now is 18289 rows & 22 columns.



AFTER CLEANING OF THE DATA FINAL OUTPUT:

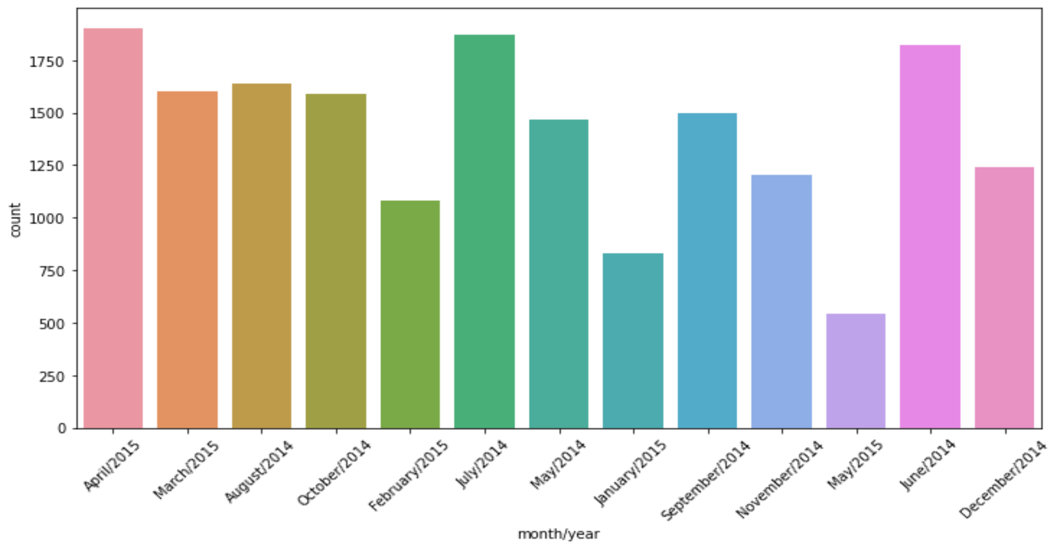
Here, I have rounded off the value of room_bath so that we can get clear outputs in our further analysis of our data, also I have removed .0 & .5 from the datapoints as they were of no use. Also, I have added additional column of month/year using dayhour variable for our analysis. So, finally our table looks like this:

TABLE:

dayhours	price	room_bed	room_bath	living_measure	lot_measure	ceil	coast	sight	condition	quality	ceil_measure	basement	yr_built	yr_renovated
2015-04-27	600000	4	2	3050	9440	1	0	0	3	8	1800	1250	1966	0
2015-03-17	190000	2	1	670	3101	1	0	0	4	6	670	0	1948	0
2014-08-20	735000	4	3	3040	2415	2	1	4	3	8	3040	0	1966	0
2014-10-10	257000	3	2	1740	3721	2	0	0	3	8	1740	0	2009	0
2015-02-18	450000	2	1	1120	4590	1	0	0	3	7	1120	0	1924	0

living_measure15	lot_measure15	furnished	total_area	lat	long	zipcode	month/year
2020	8660	0	12490	47.7228	-122.183	98034	April/2015
1660	4100	0	3771	47.5546	-122.274	98118	March/2015
2620	2433	0	5455	47.5188	-122.256	98118	August/2014
2030	3794	0	5461	47.3363	-122.213	98002	October/2014
1120	5100	0	5710	47.5663	-122.285	98118	February/2015

Analysis of dayhours



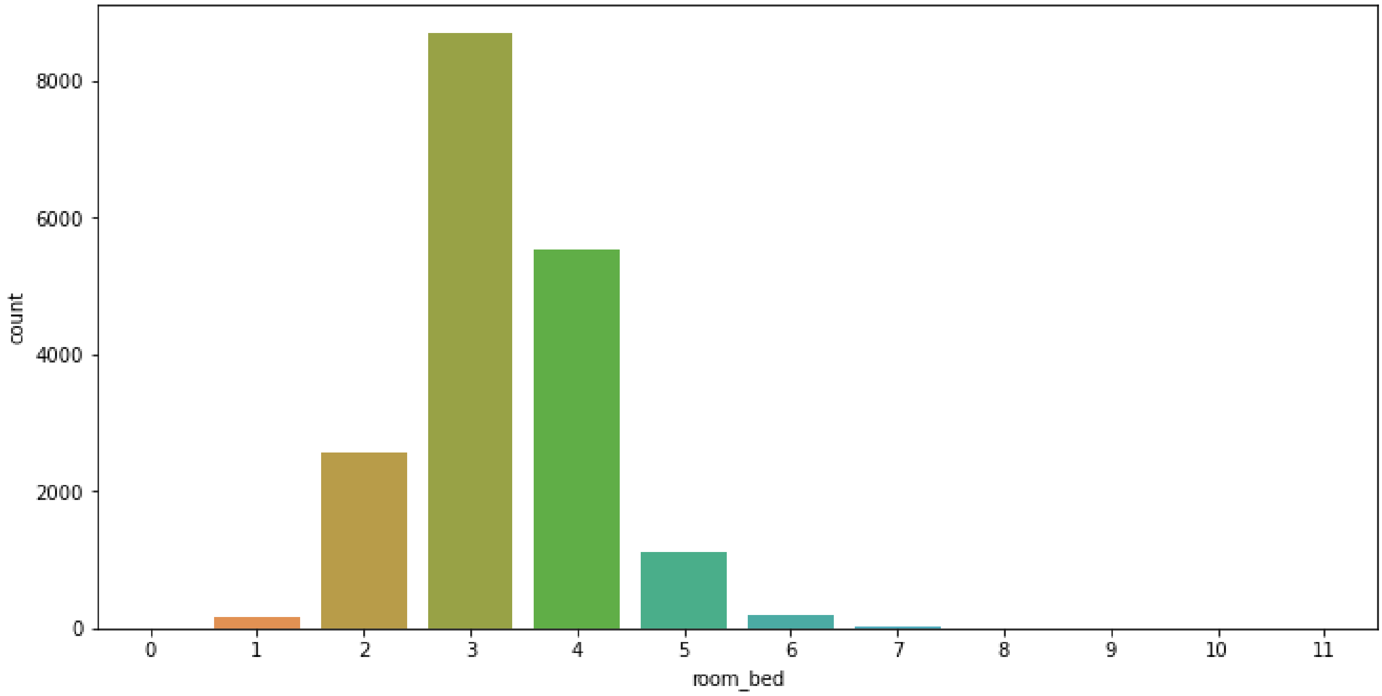
From the above, we can conclude that in april 2015 and june and july of 2014 most houses are sold.

```
month/year
April/2015          507327.633018
August/2014         480635.503360
December/2014       469974.957224
February/2015       462635.759704
January/2015        465124.644150
July/2014           491450.990928
June/2014           501607.509341
March/2015          499022.900249
May/2014            492102.350614
May/2015            502737.100917
November/2014       467927.724252
October/2014        478020.134047
September/2014      478270.602804
```

April month have the highest mean price in the time line of the sales of the propertie s is from May-2014 to May-2015.

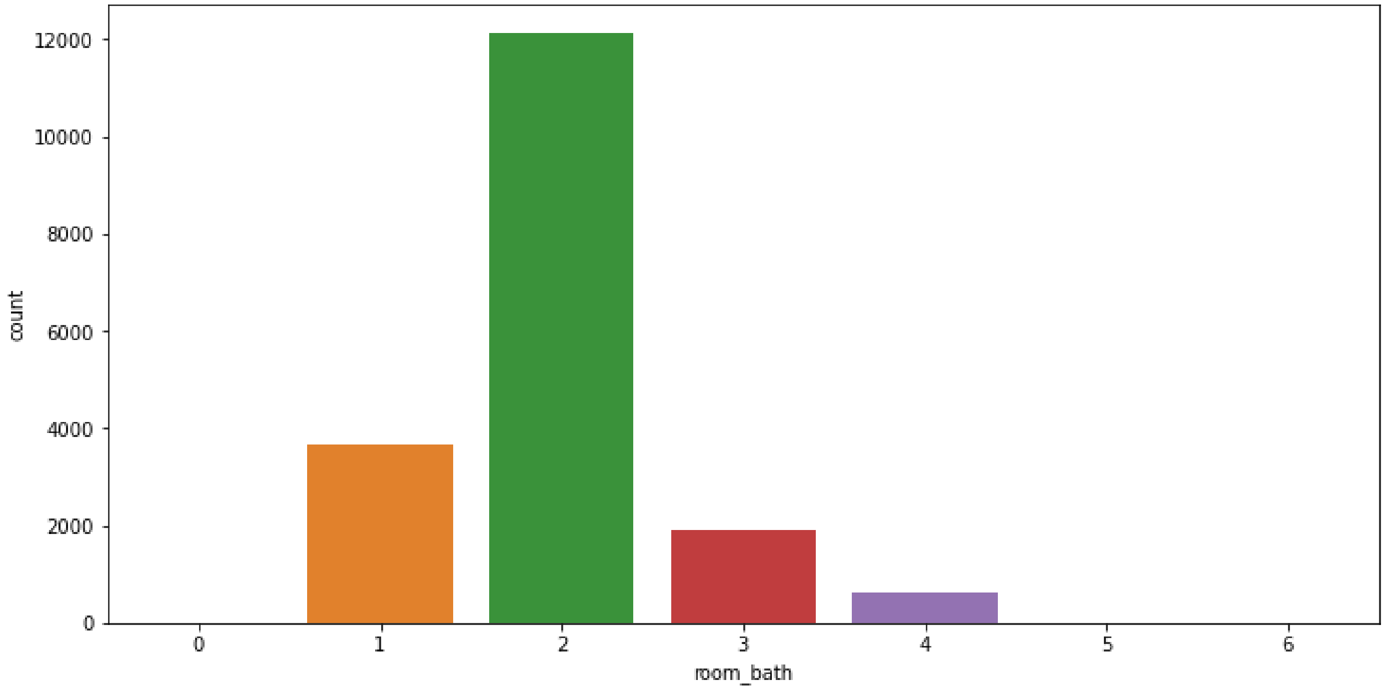
Univariat Analysis of each column

Analysis of room_bed



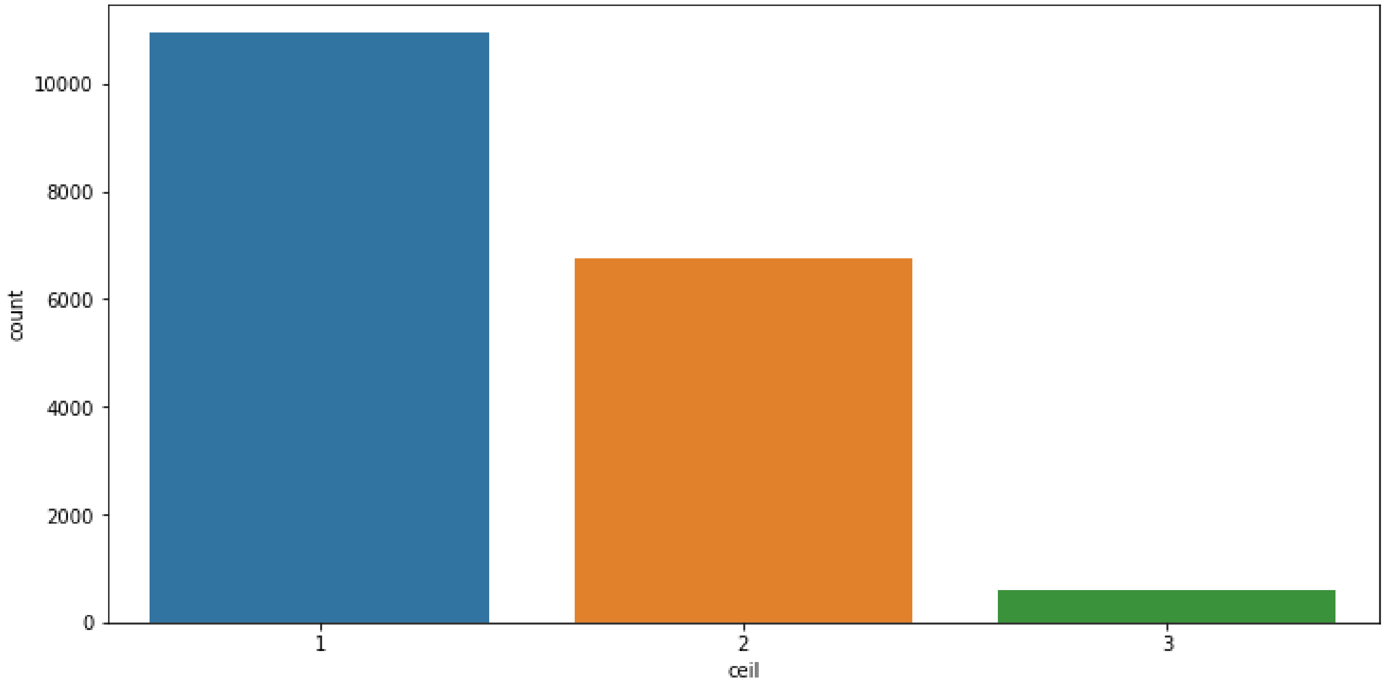
Most of the houses have 3 or 4 bedrooms.

Analysis of room_bath



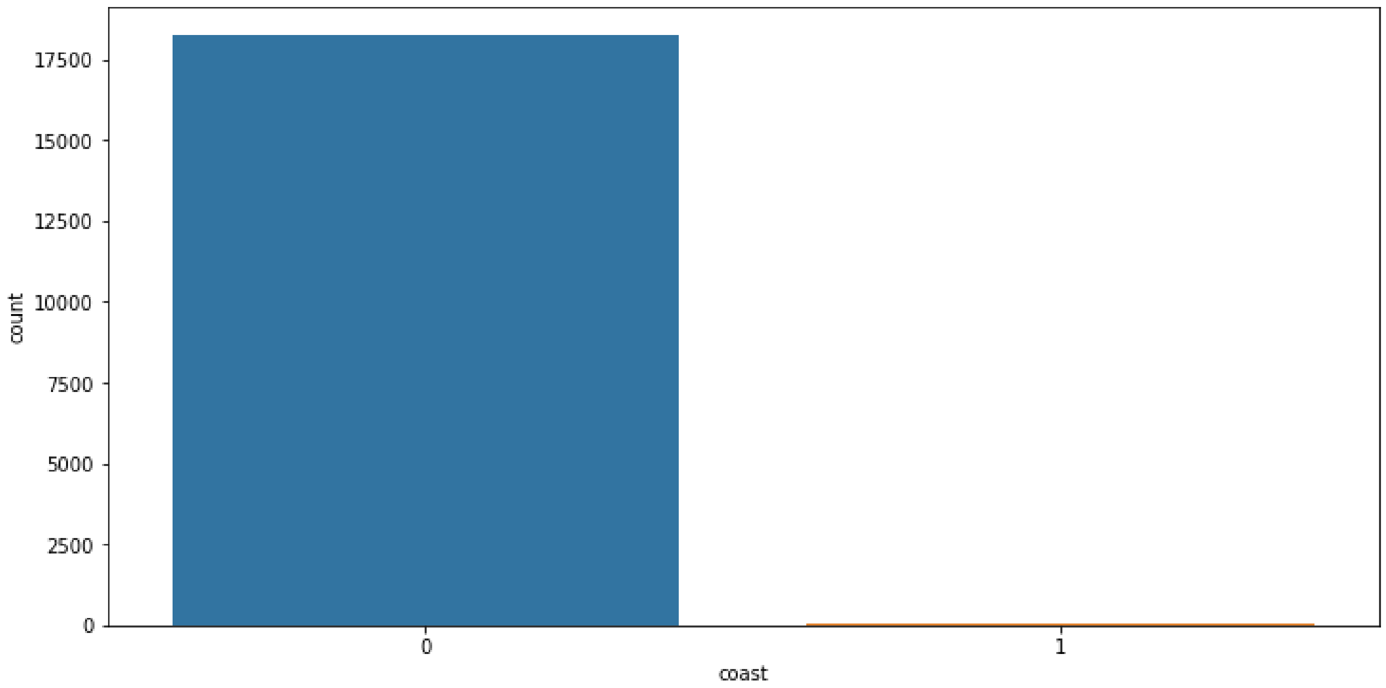
Majority of the houses have 2 bathroom's followed by 1 & 3.

CEIL



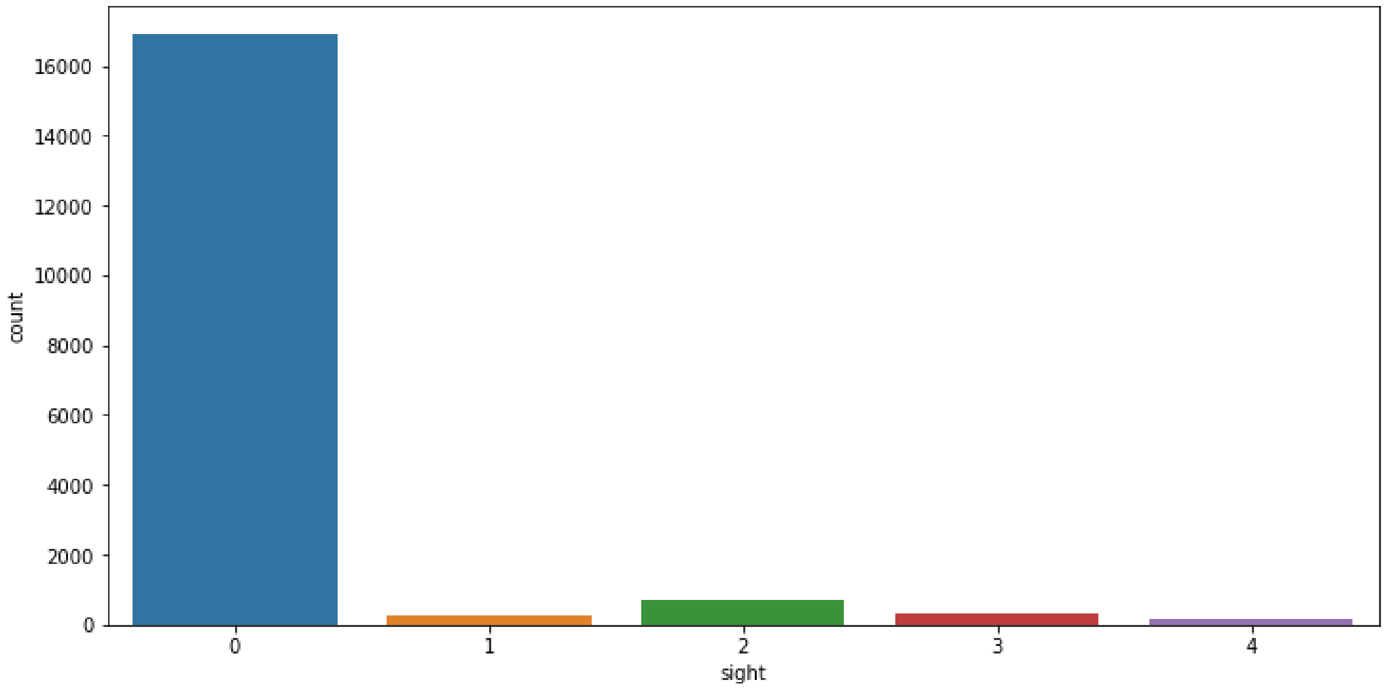
Most houses have 1 and 2 floors

Coast



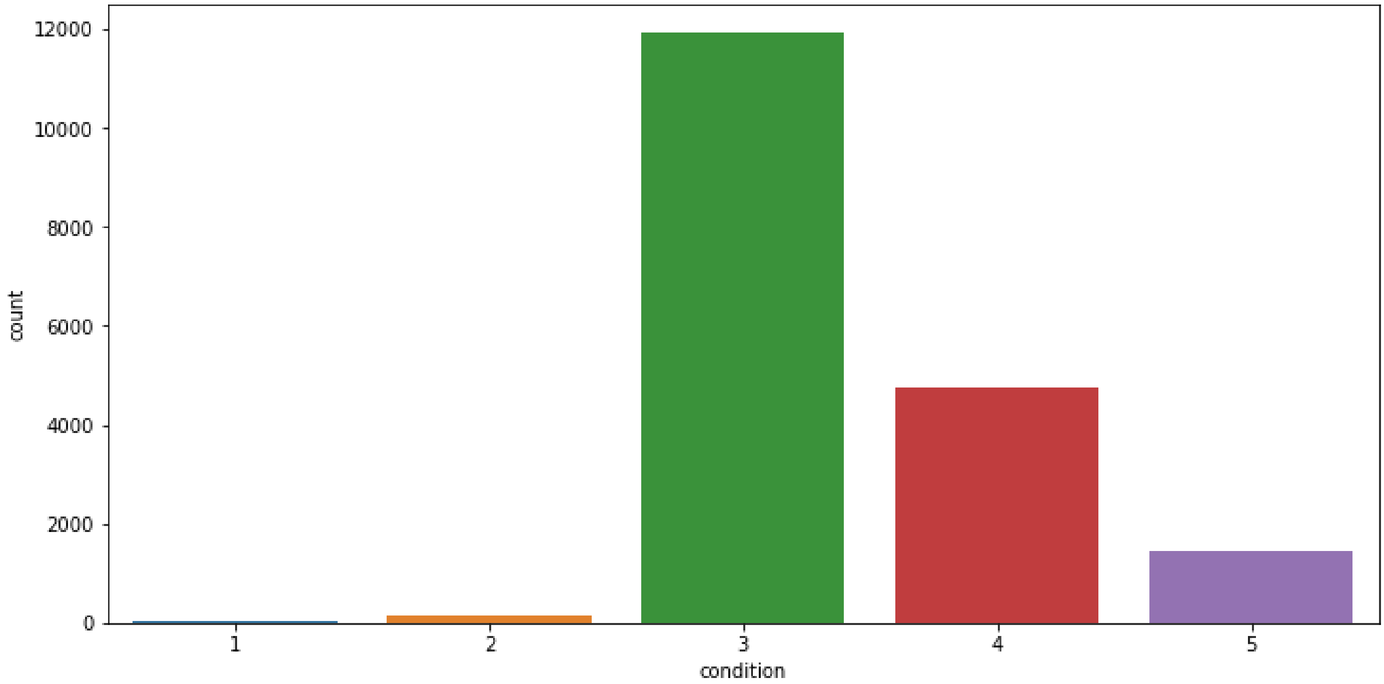
Most houses don't have waterfront view, very few are waterfront.

Sight



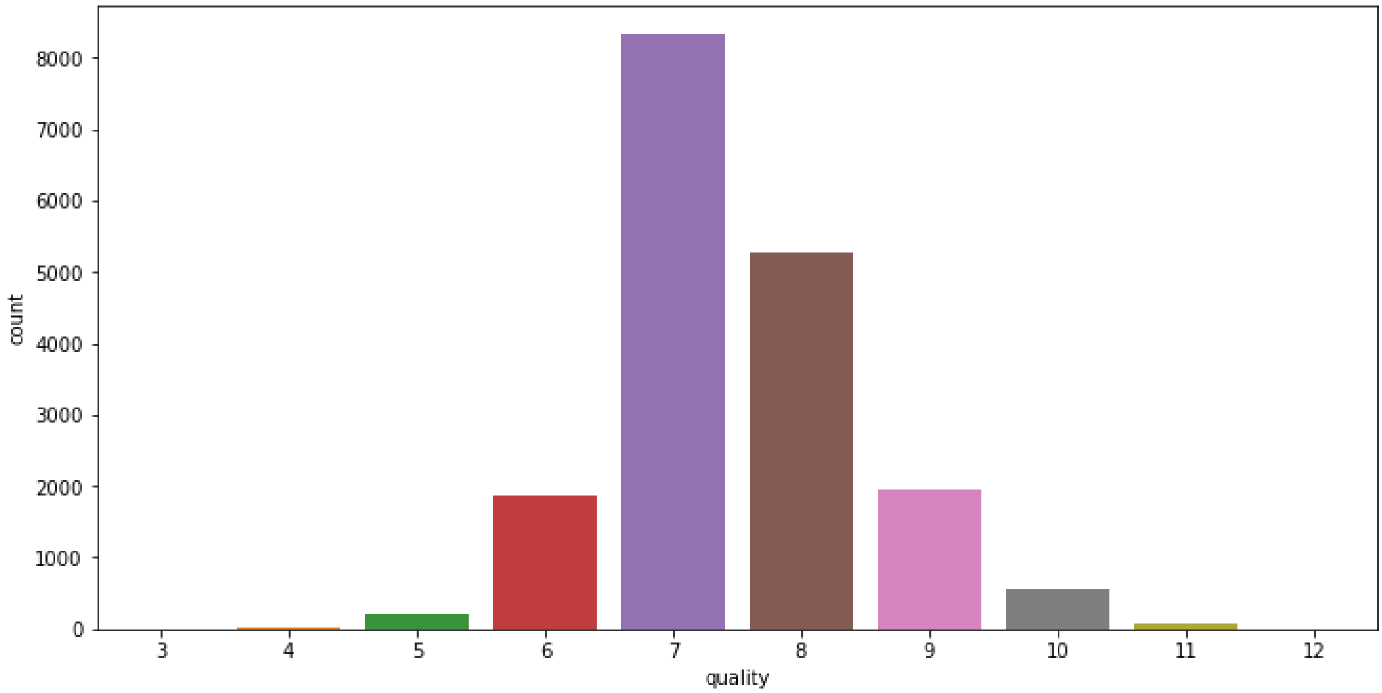
Most sights have not been viewed.

Condition



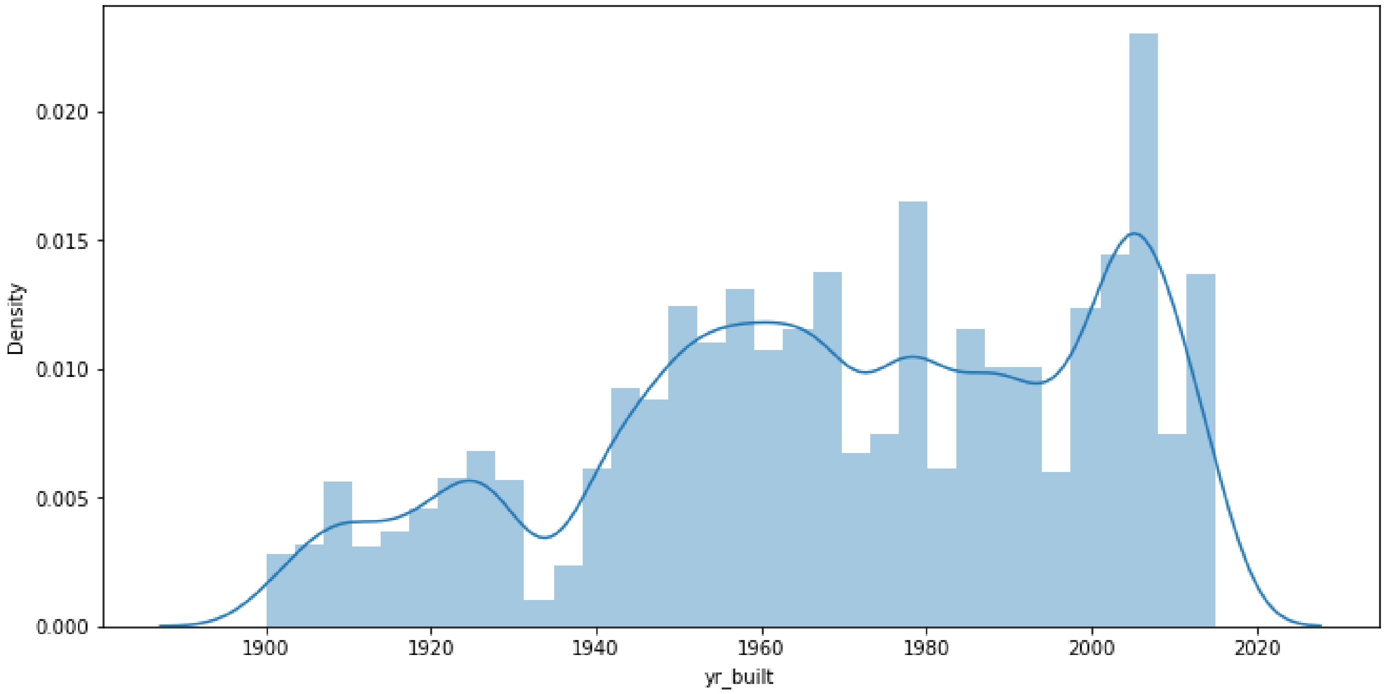
Overall most houses are rated as 3 and above for its condition overall.

Quality



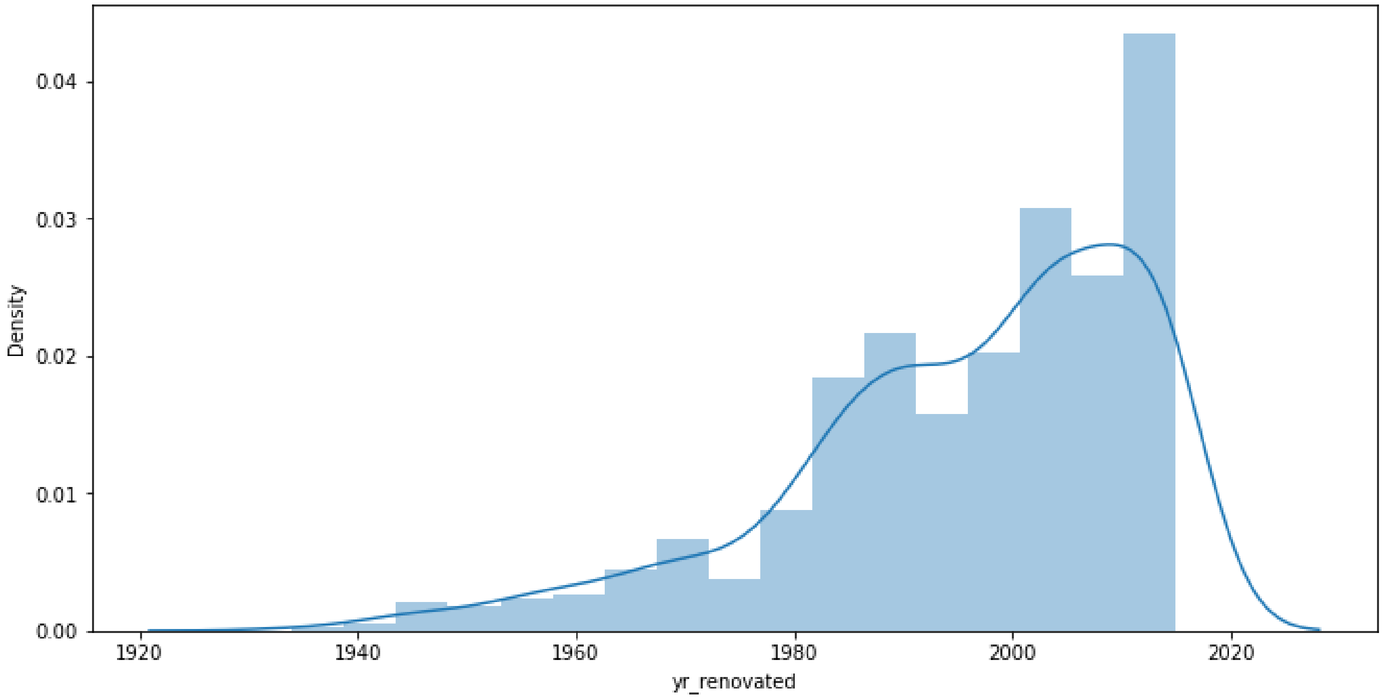
Most of the housing unit have been given grade 7 followed by 8 & 9.

Analysis of yr_built



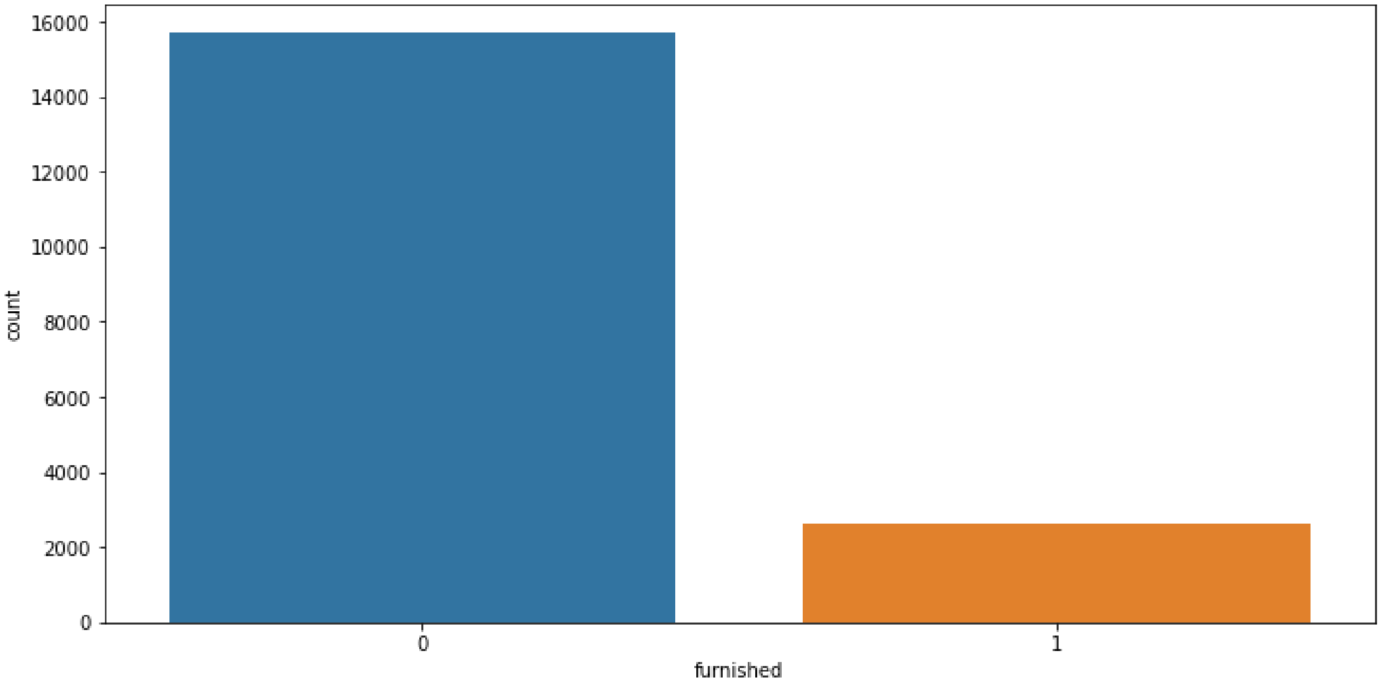
The built year of the properties range from 1900 to 2014 and we can see upward trend with time.

Analysis of yr_renovated



There is an upward trend in renovation's continuing from 1980.

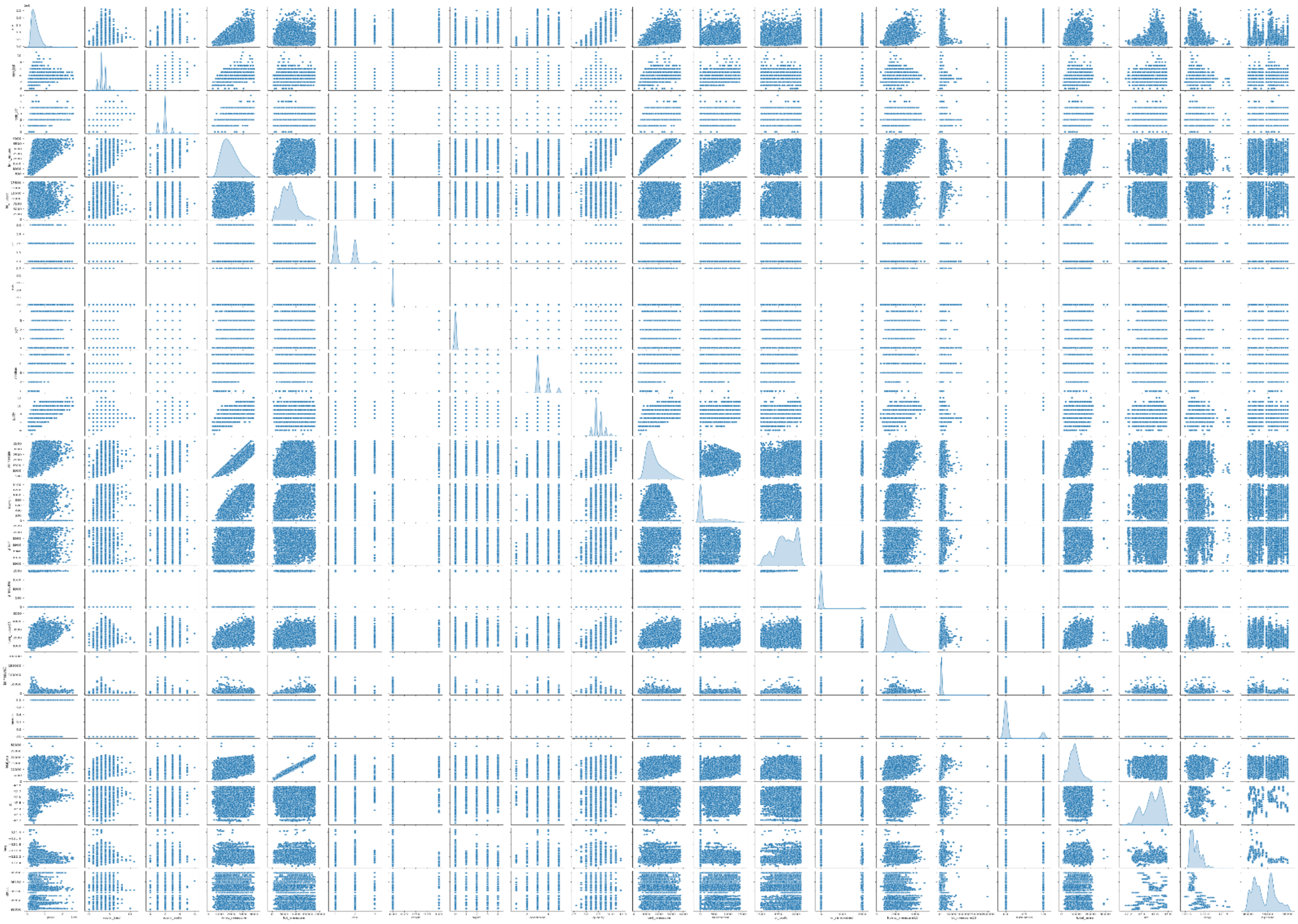
Furnished



Most properties are not furnished.

Bi-Variate Analysis:

Pairplot:



price: price distribution is Right-Skewed as we deduced earlier from our 5-factor analysis

room_bed: our target variable (price) and room_bed plot is not linear. It's distribution have lot of gaussians

room_bath: It's plot with price has somewhat linear relationship. Distribution has number of gaussians.

living_measure: Plot against price has strong linear relationship. It also have linear relationship with room_bath variable. So might remove one of these 2. Distribution is Right-Skewed.

lot_measure: No clear relationship with price.

ceil: No clear relationship with price. We can see, it's have 6 unique values only. Therefore, we can convert this column into categorical column for values.

coast: No clear relationship with price. Clearly it's categorical variable with 2 unique values.

sight: No clear relationship with price. This has 5 unique values. Can be converted to Categorical variable.

condition: No clear relationship with price. This has 5 unique values. Can be converted to Categorical variable.

quality: Somewhat linear relationship with price. Has discrete values from 1 - 13. Can be converted to Categorical variable.

ceil_measure: Strong linear relationship with price. Also with room_bath and living_measure features. Distribution is Right-Skewed.

basement: No clear relationship with price.

yr_built: No clear relationship with price.

yr_renovated: No clear relationship with price. Have 2 unique values. Can be converted to Categorical Variable which tells whether house is renovated or not.

zipcode, lat, long: No clear relationship with price or any other feature.

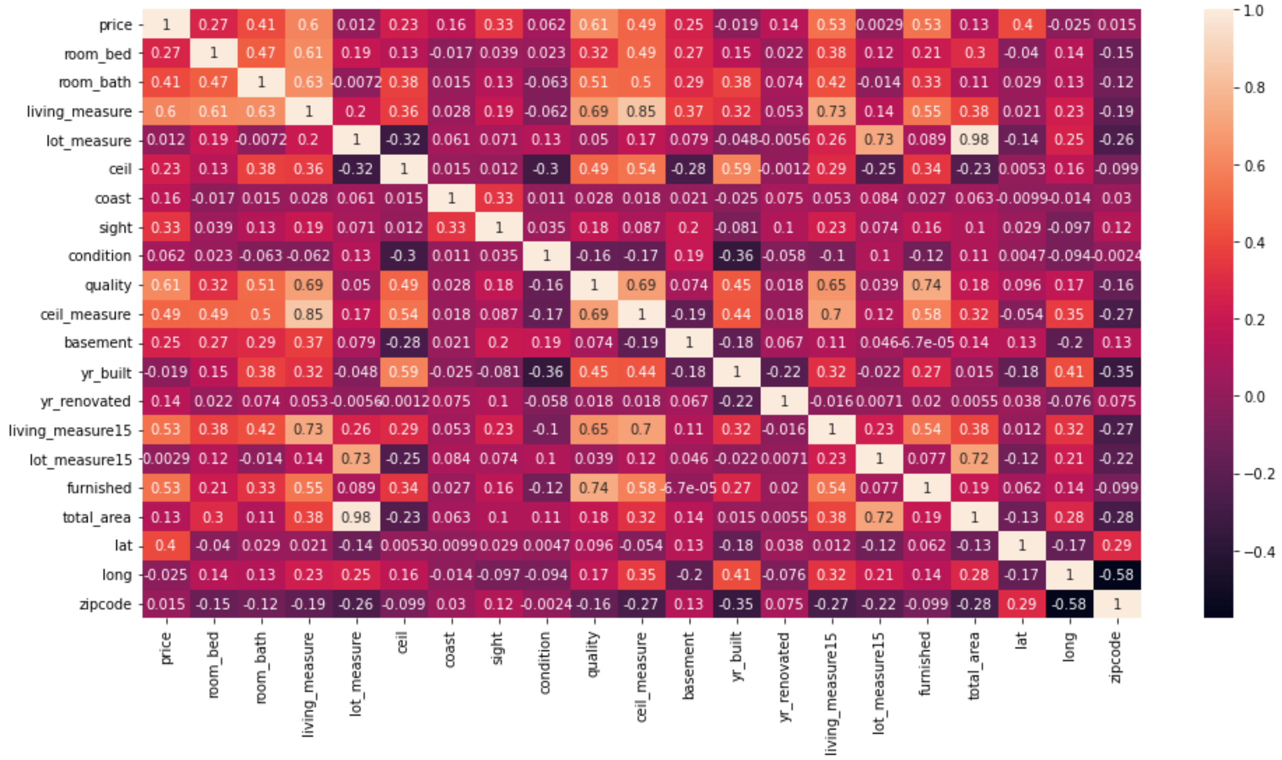
living_measure15: Somewhat linear relationship with target feature. It's same as living_measure. Therefore we can drop this variable.

lot_measure15: No clear relationship with price or any other feature.

furnished: No clear relationship with price or any other feature. 2 unique values so can be converted to Categorical Variable

total_area: No clear relationship with price. But it has Very Strong linear relationship with lot_measure. So one of it can be dropped.

HeatMap :



We have linear relationships in below featues as we got to know from above matrix

1.price: room_bath, living_measure, quality, living_measure15, furnished

2.living_measure: price, room_bath. So we can consider dropping 'room_bath' variable.

3.quality: price, room_bath, living_measure

4.ceil_measure: price, room_bath, living_measure, quality

5.living_measure15: price, living_measure, quality. So we can consider dropping living_measure15 as well. As it's giving same info as living_measure.

6.lot_measure15: lot_measure. Therefore, we can consider dropping lot_measure15, as it's giving same info.

7.furnished: quality

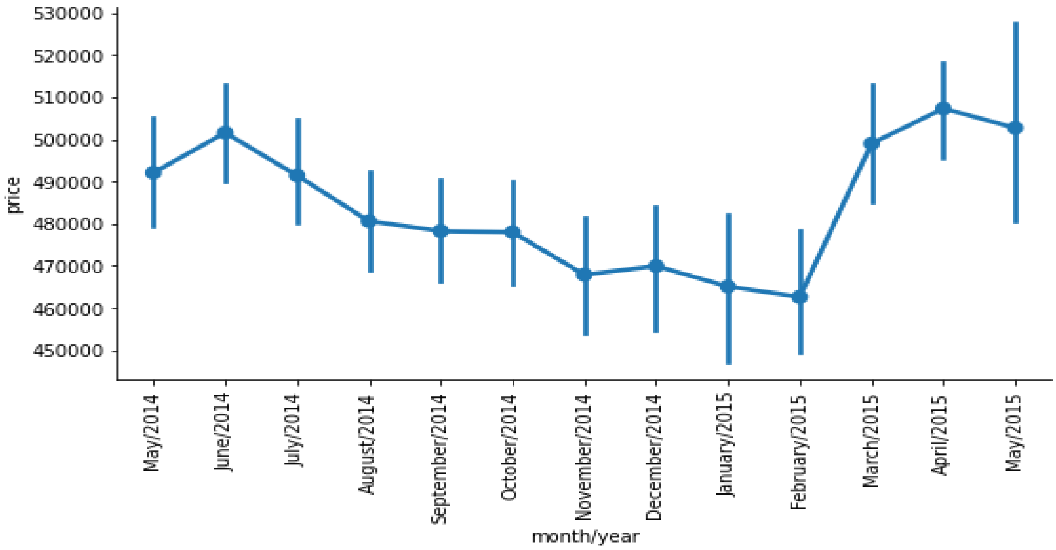
8.total_area: lot_measure, lot_measure15. Therefore, we can consider dropping total_area feature as well. As it's giving same info as lot_measure.

Bivariate Analysis of Variables

month_year

	mean	median	size
month/year			
April/2015	507327.633018	450000.0	1902
August/2014	480635.503360	420000.0	1637
December/2014	469974.957224	406000.0	1239
February/2015	462635.759704	406375.0	1082
January/2015	465124.644150	400000.0	829

	mean	median	size
month/year			
July/2014	491450.990928	438500.0	1874
June/2014	501607.509341	441000.0	1820
March/2015	499022.900249	432625.0	1604
May/2014	492102.350614	435555.0	1466
May/2015	502737.100917	440000.0	545
November/2014	467927.724252	415000.0	1204
October/2014	478020.134047	422500.0	1589
September/2014	478270.602804	427500.0	1498

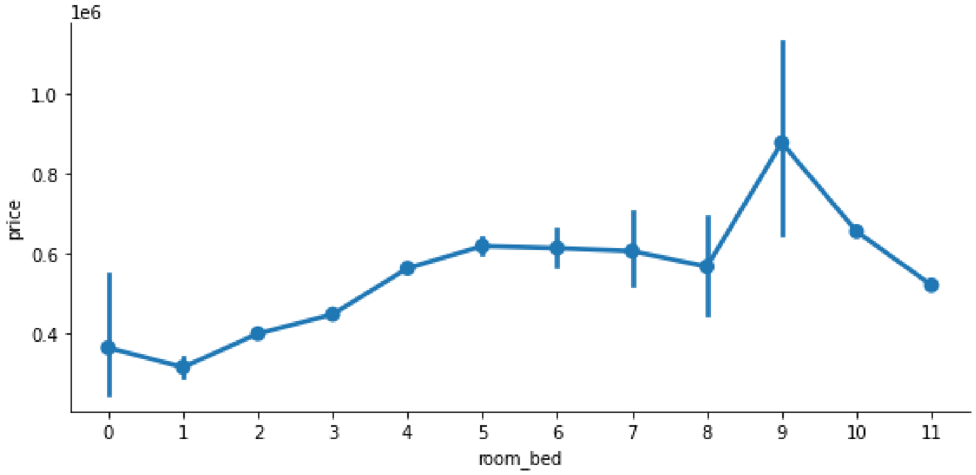


The mean price of the houses tend to be high during March,April, May as compared to that of September, October, November,December period.

room_bed

	mean	median	size
room_bed			
0	362590.000000	304000.0	10

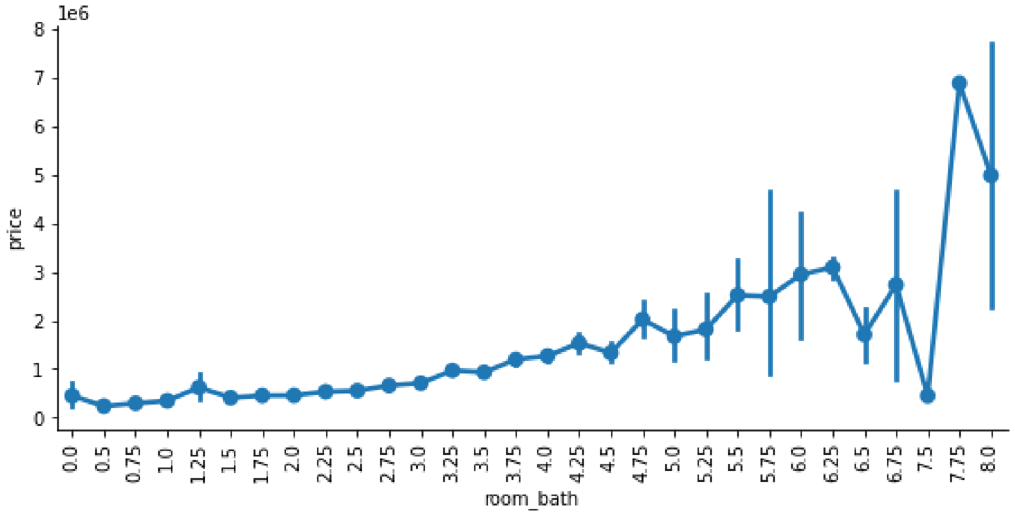
	mean	median	size
room_bed			
1	313286.847059	297000.0	170
2	397850.747467	375000.0	2566
3	446672.407075	400000.0	8679
4	562585.007401	505000.0	5540
5	618386.443643	545000.0	1109
6	613235.770950	585444.0	179
7	605732.590909	577500.0	22
8	566571.428571	575000.0	7
9	878499.750000	817000.0	4
10	655000.000000	655000.0	2
11	520000.000000	520000.0	1



There is clear increasing trend in price with room_bed.

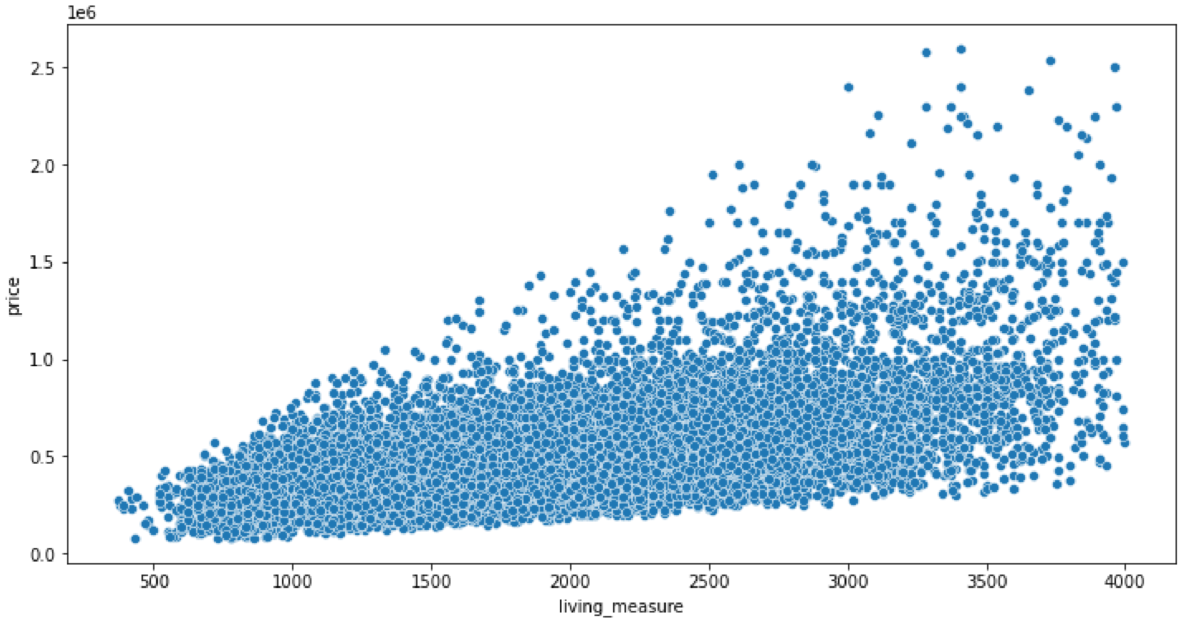
room_bath

	mean	median	size
room_bath			
0	3.510500e+05	273000.0	9
1	3.479121e+05	320000.0	3647
2	4.830434e+05	436000.0	12118
3	6.567162e+05	595000.0	1895
4	8.354890e+05	775000.0	610
5	1.019611e+06	643500.0	9
6	5.400000e+05	540000.0	1



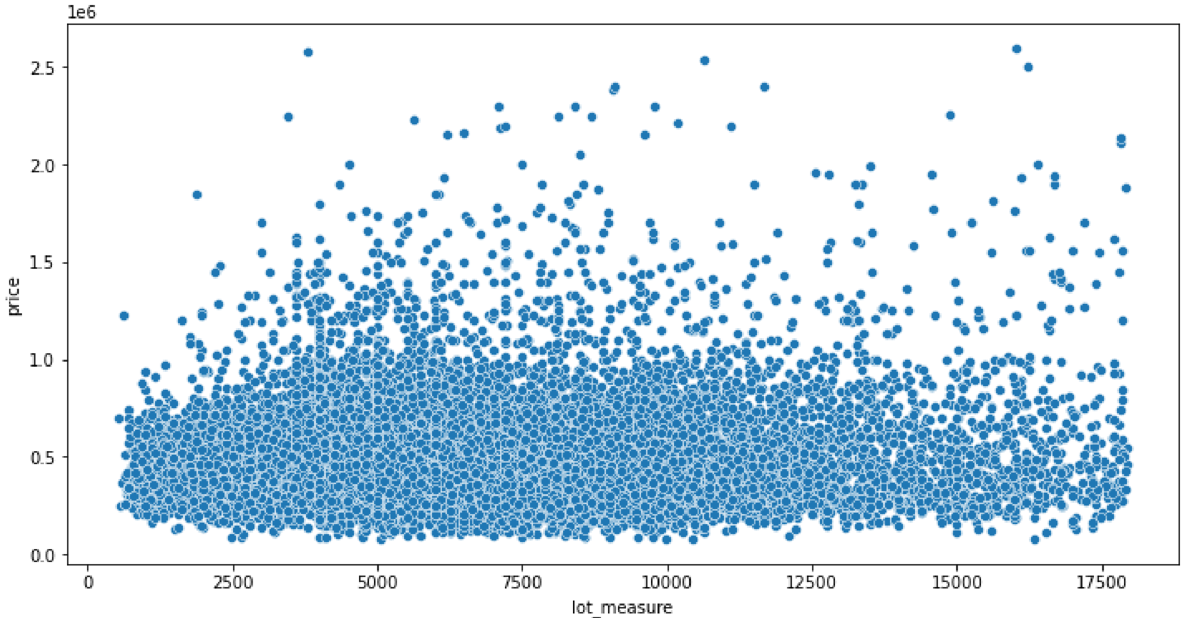
Overall mean and median price increares with increasing room_bath there is upward trend in price with increase in room_bath.

living_measure



There is clear increment in price of the property with increment in the living measure.

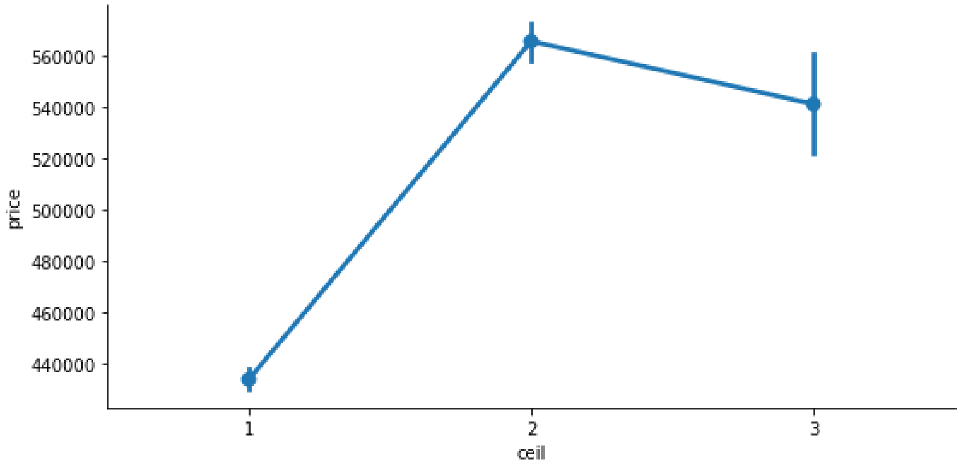
lot_measure



The figure is not showing any trend it could mean that there is little to no relationship between price and lot_measure.Almost 95% of the houses have <17950 lot_measure.

Ceil

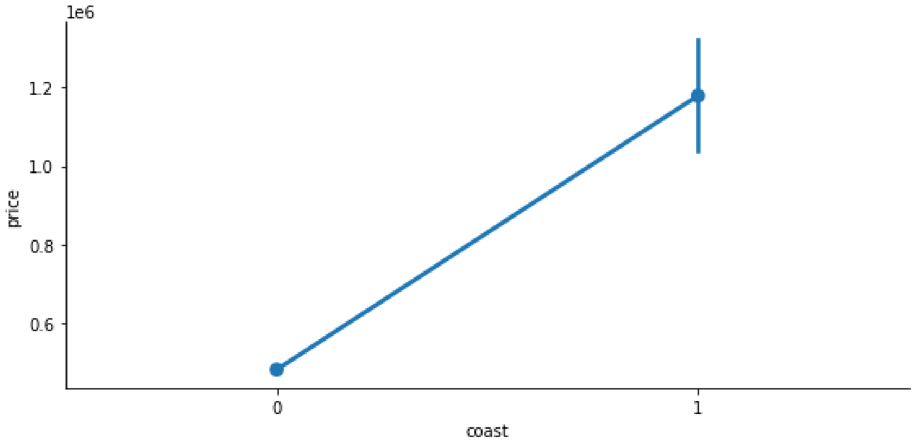
	mean	median	size
ceil			
1	433861.412561	393000.0	10939
2	565689.601272	496000.0	6759
3	541201.898477	481000.0	591



Initially the price is increasing and after that we can see a slight fall in price as it goes further.

Coast

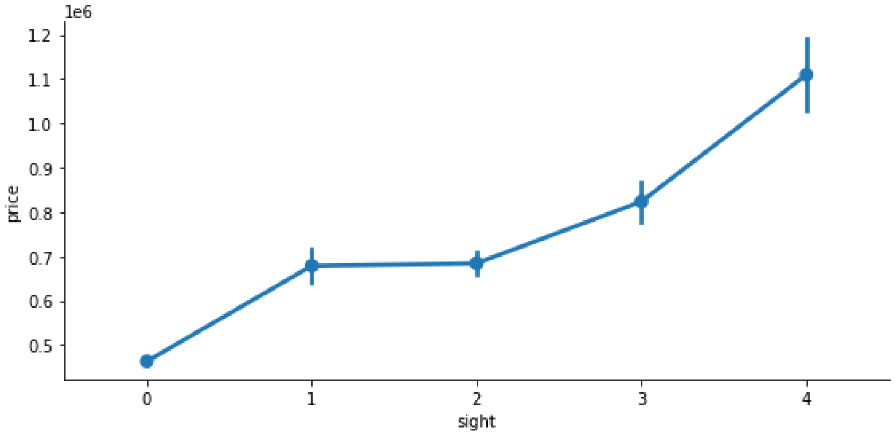
	living_measure		price	
	median	**mean**	**median**	**mean**
coast				
0	1800.0	1898.520108	428000.0	4.836974e+05
1	2165.0	2235.500000	1125000.0	1.177483e+06



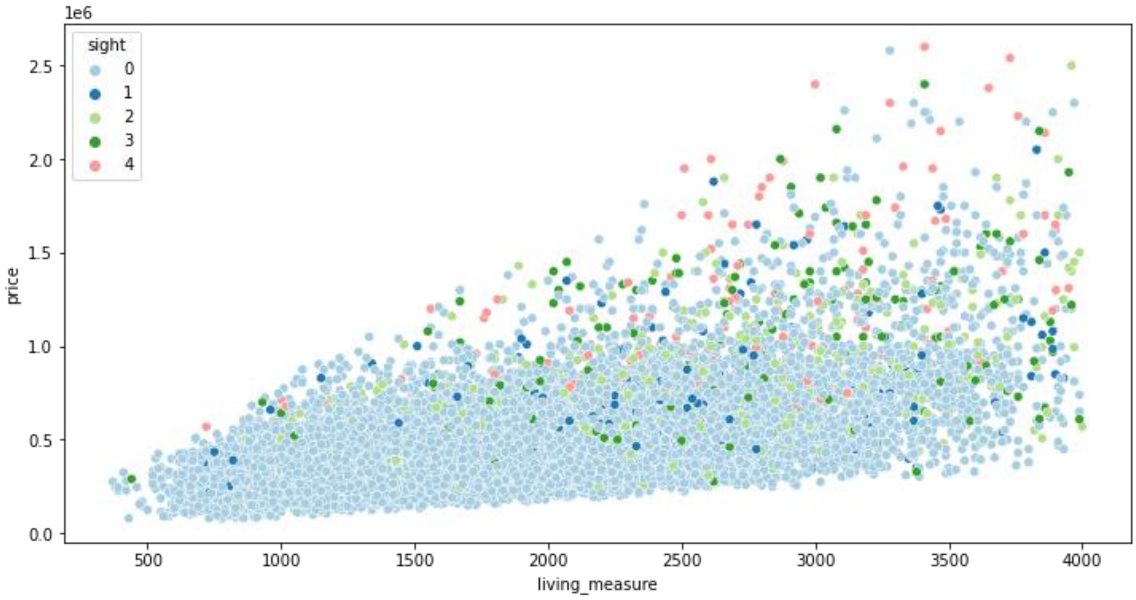
Mean and median of waterfront view is high however such houses are very small in compare to non-waterfront Also, living_measure mean and median is greater for waterfront house. The house properties with water_front tend to have higher price compared to that of non-water_front properties.

Sight

	price			living_measure		
	mean	median	size	mean	median	size
sight						
0	4.635073e+05	415000.0	16890	1862.120604	1770.0	16890
1	6.788832e+05	649975.0	266	2263.500000	2245.0	266
2	6.846175e+05	635000.0	689	2276.008708	2240.0	689
3	8.239130e+05	720000.0	295	2517.647458	2500.0	295
4	1.109926e+06	975000.0	149	2541.899329	2610.0	149



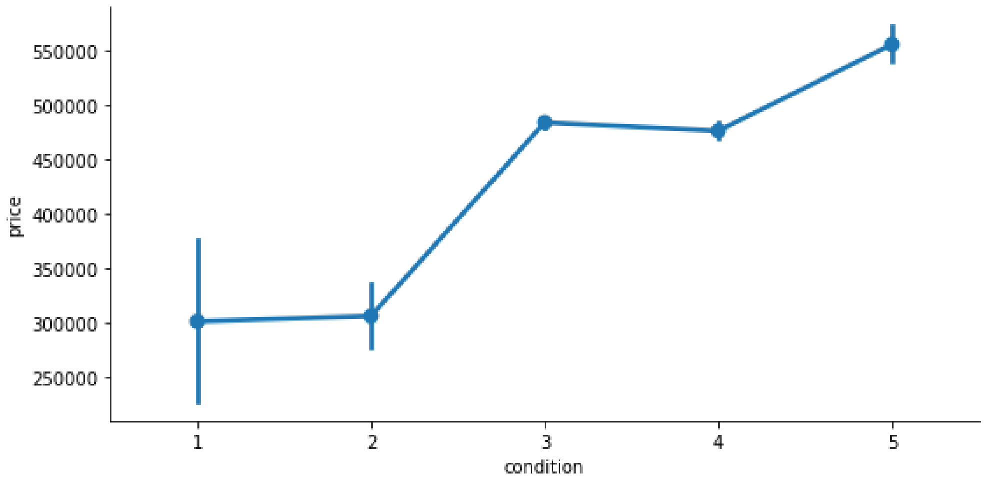
The house sighted more have high price (mean and median) and have large living area as well. Properties with higher price have more no.of sights compared to that of houses with lower price.



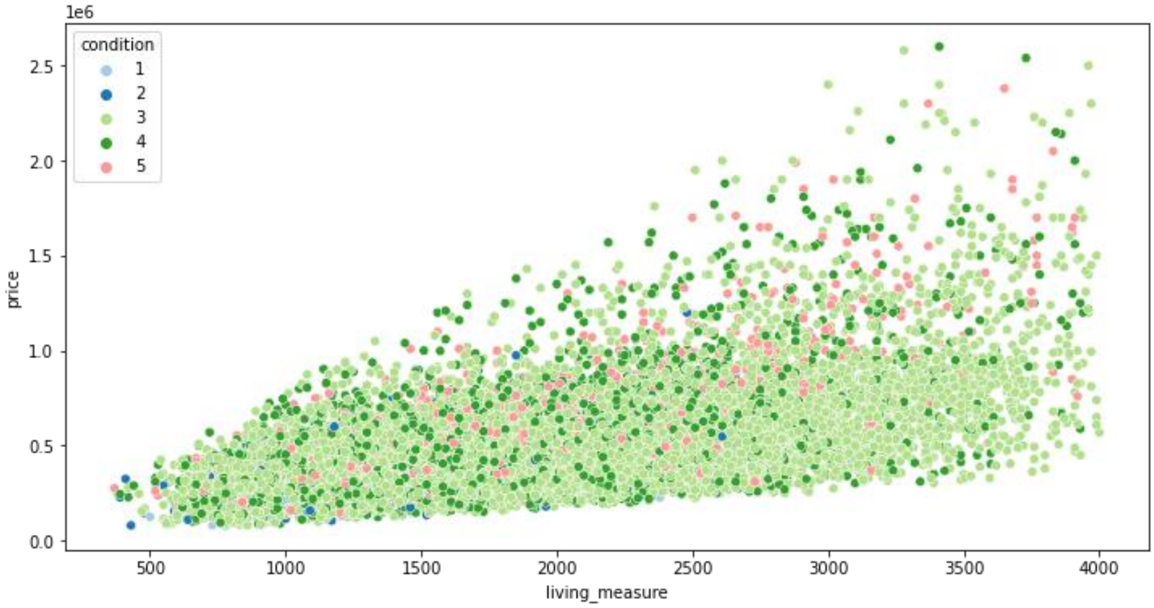
Viewed in relation with price and living_measure Costlier houses with large living area are sighted more. The above graph also justify that: Properties with higher price have more no.of sights compared to that of houses with lower price.

Condition

	price			living_measure		
	mean	median	size	mean	median	size
condition						
1	301235.714286	255000.0	21	1231.428571	1010.0	21
2	306112.679104	270630.0	134	1315.522388	1235.0	134
3	483820.057500	428000.0	11913	1955.198523	1850.0	11913
4	476142.186120	420000.0	4755	1790.619558	1730.0	4755
5	555393.478854	500000.0	1466	1865.014325	1800.0	1466



As the condition rating increases its price and living measure mean and median also increases. The price of the house increases with condition rating of the house.

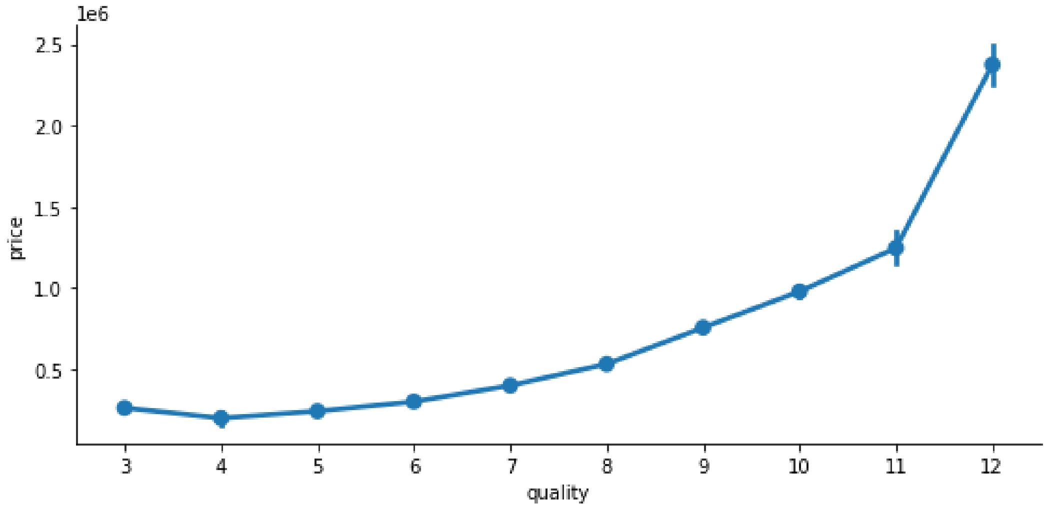


Viewed in relation with price and living_measure. Most houses are rated as 3 or more. So we found out that smaller houses are in better condition and better condition houses are having higher prices.

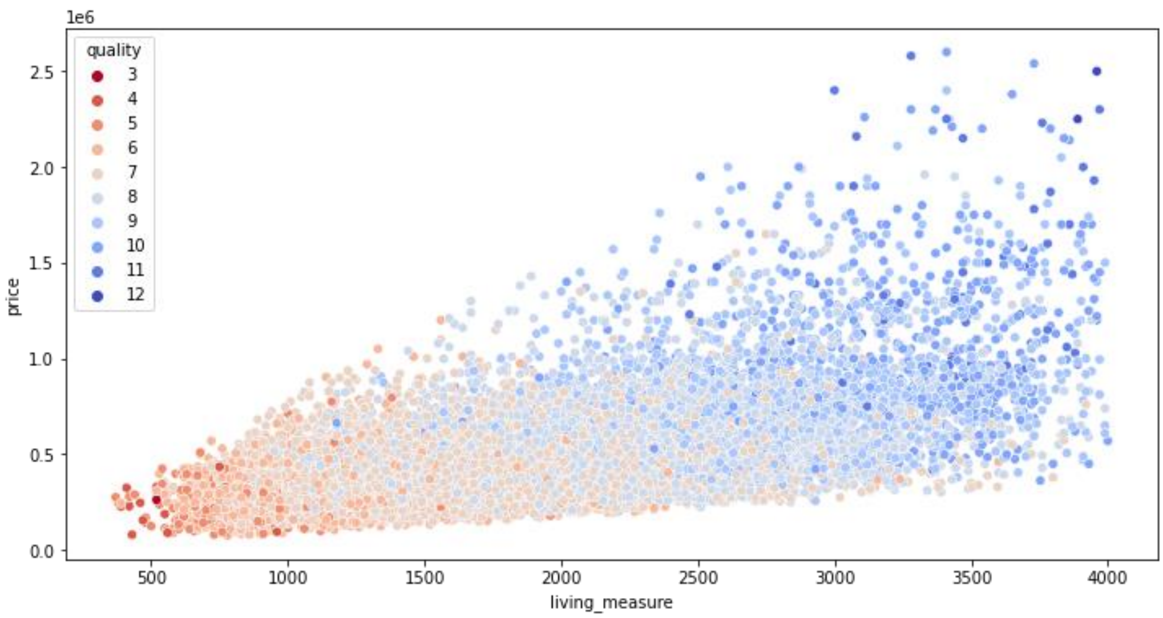
Quality

	price			living_measure		
	mean	**median**	**size**	**mean**	**median**	**size**
quality						
3	2.620000e+05	262000.0	1	520.000000	520.0	1
4	1.995262e+05	188000.0	21	601.904762	560.0	21
5	2.418038e+05	225000.0	196	951.591837	855.0	196
6	3.006807e+05	275000.0	1869	1173.516854	1100.0	1869
7	4.002021e+05	371500.0	8325	1662.533093	1610.0	8325
8	5.324621e+05	500000.0	5276	2113.186315	2090.0	5276
9	7.563580e+05	715000.0	1954	2723.395087	2735.0	1954
10	9.787350e+05	858250.0	568	3115.123239	3180.0	568

	price			living_measure		
	mean	median	size	mean	median	size
quality						
11	1.246639e+06	1090000.0	77	3395.311688	3450.0	77
12	2.375000e+06	2375000.0	2	3925.000000	3925.0	2

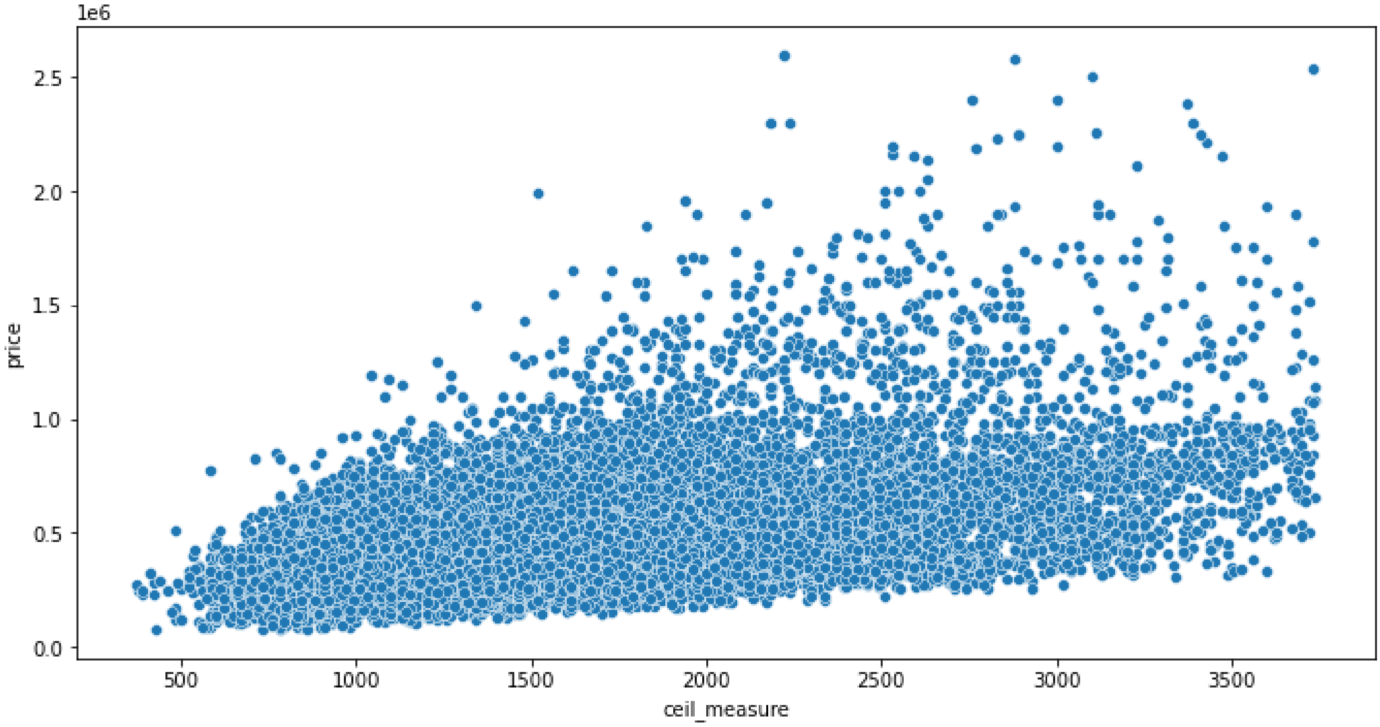


There is clear increase in price of the house with higher rating on quality.



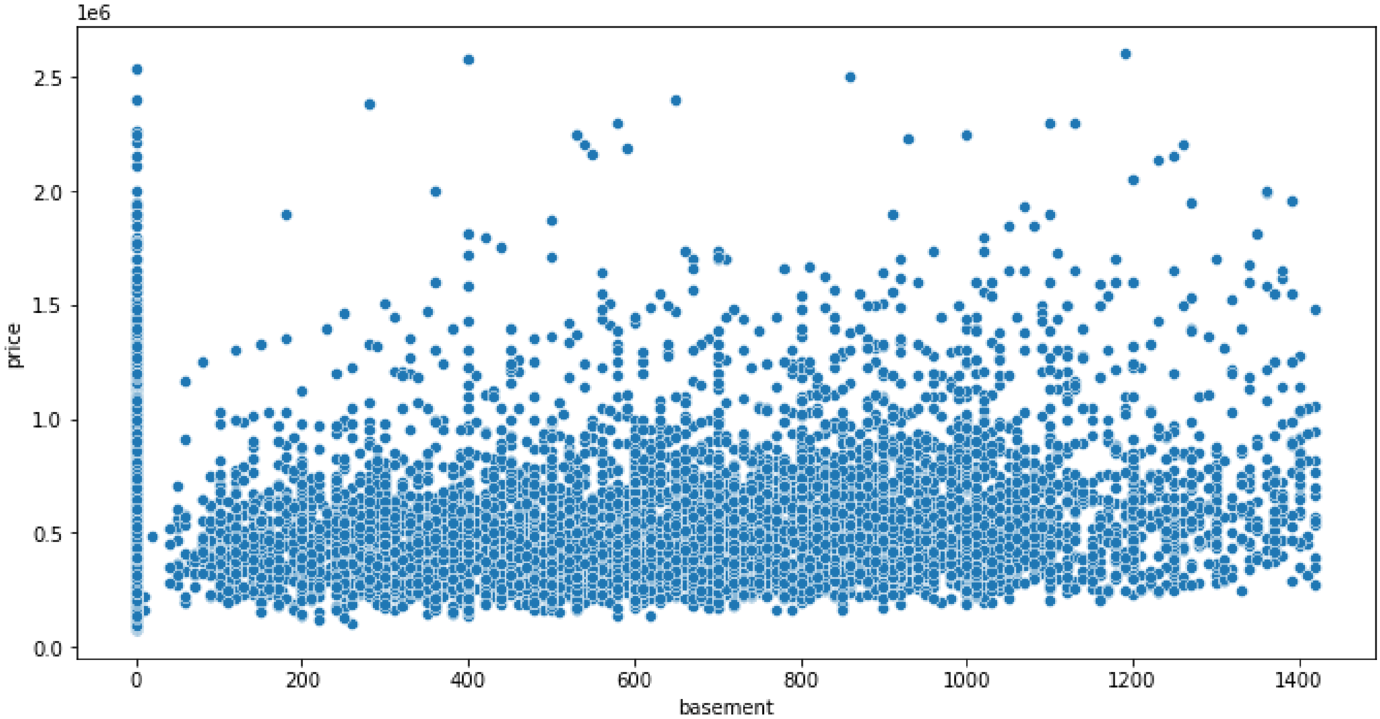
Viewed in relation with price and living_measure. Most houses are graded as 6 or more.

ceil_measure



There is upward trend in price with ceil_measure.

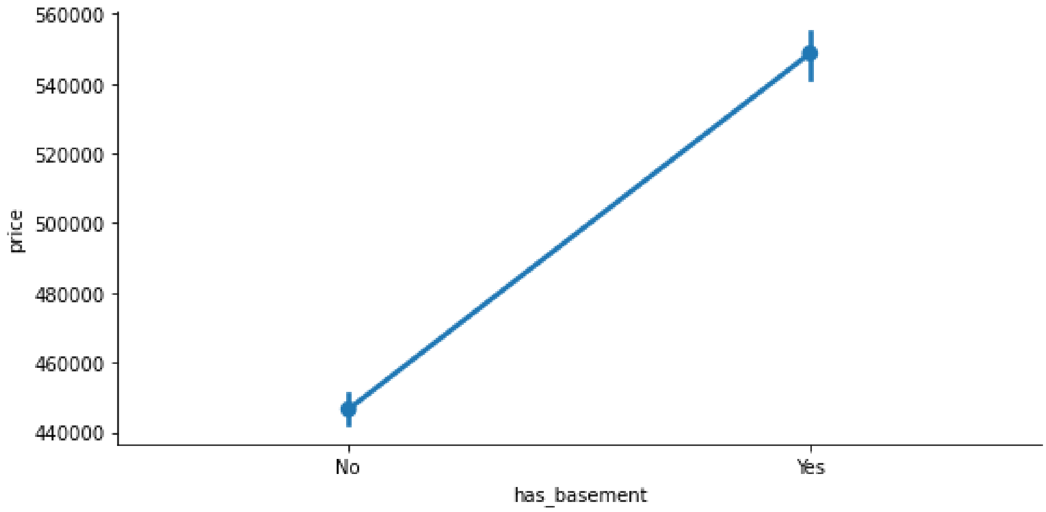
Basement



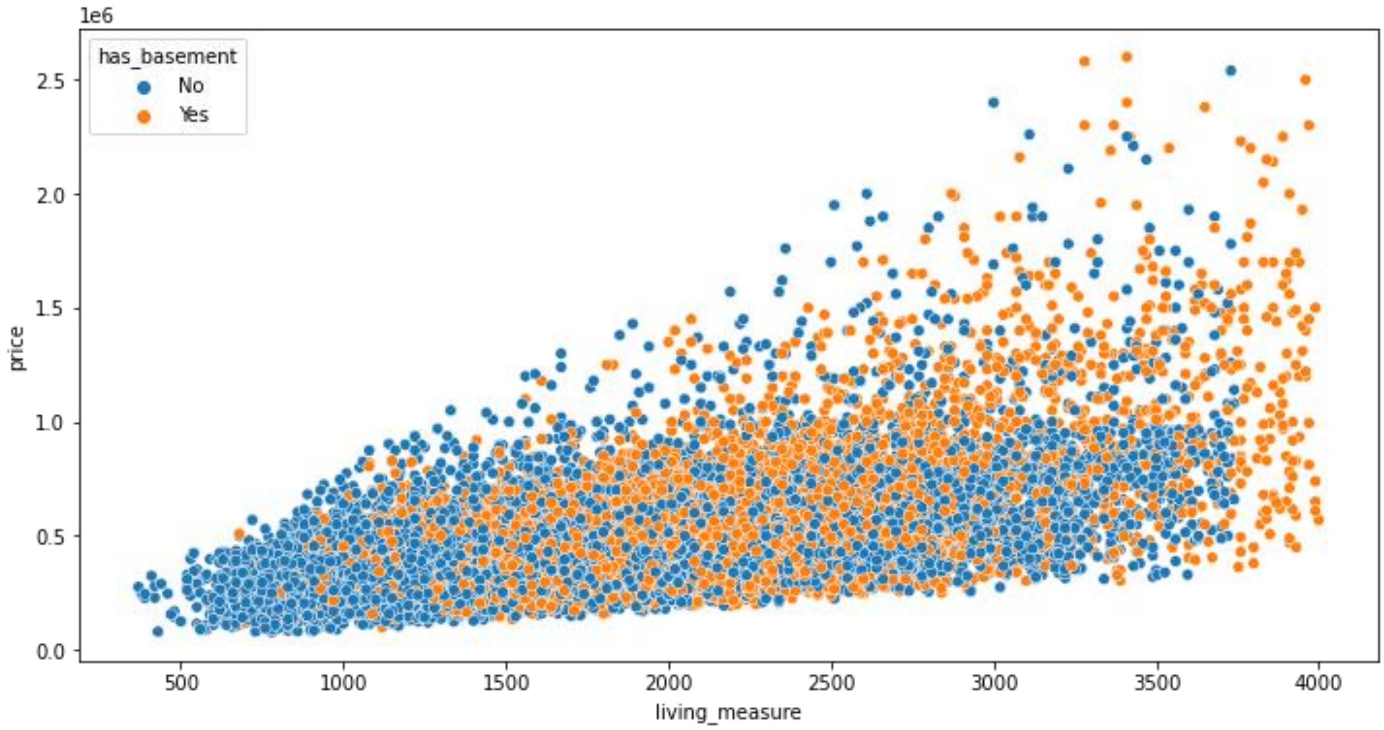
We will create the categorical variable for basement 'has_basement' for houses with basement and no basement.This categorical variable will be used for further analysis. Price increases with increase in ceil measure.

Adding a new caegorical variable for looking into the data which houses are having basements and which are not

	price			living_measure		
	mean	median	size	mean	median	size
has_basement						
No	446607.035921	390000.0	11219	1788.872449	1650.0	11219
Yes	548638.188543	490000.0	7070	2075.469165	1990.0	7070



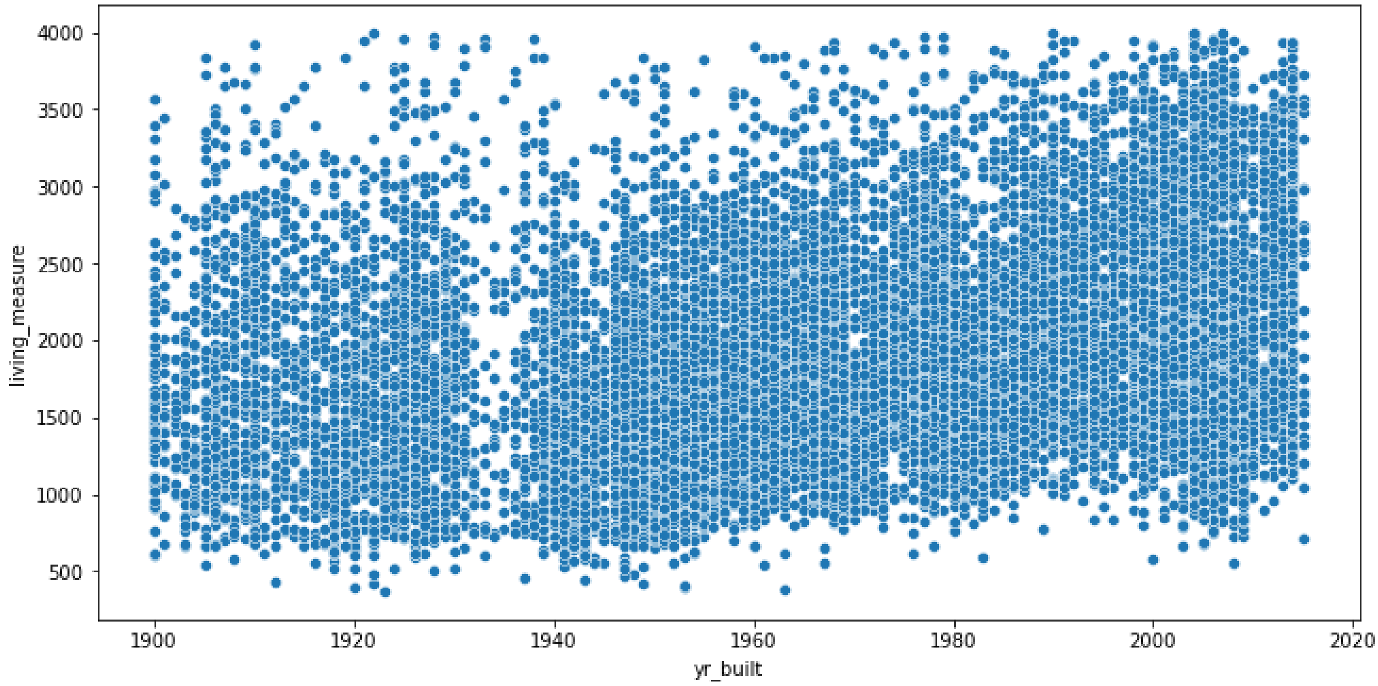
After binning we data shows with basement houses are costlier and have higher Prices. The houses with basement has better price compared to that of houses without basement



Houses having basement have higher price & living measure.

Yr_built

	Mean	median	size
yr_built			
1900	590996.135802	550000.0	81
1901	557108.344828	550000.0	29
1902	620848.000000	591000.0	25
1903	484705.500000	461000.0	44
1904	527791.837209	478000.0	43
...	...	...	...
2011	515572.644628	430000.0	121
2012	507776.937888	425000.0	161
2013	563532.751445	505000.0	173
2014	625021.102161	565997.0	509
2015	667531.533333	605805.5	30



We will create new variable: Houselandratio - This is proportion of living area in the total area of the house. We will explore the trend of price against this houselandratio.

Creating a new column for calculating the percentage of living space in the house

house_land_ratio

```
4886      8.0
9357     17.0
1635     42.0
17531    55.0
17530    34.0
```

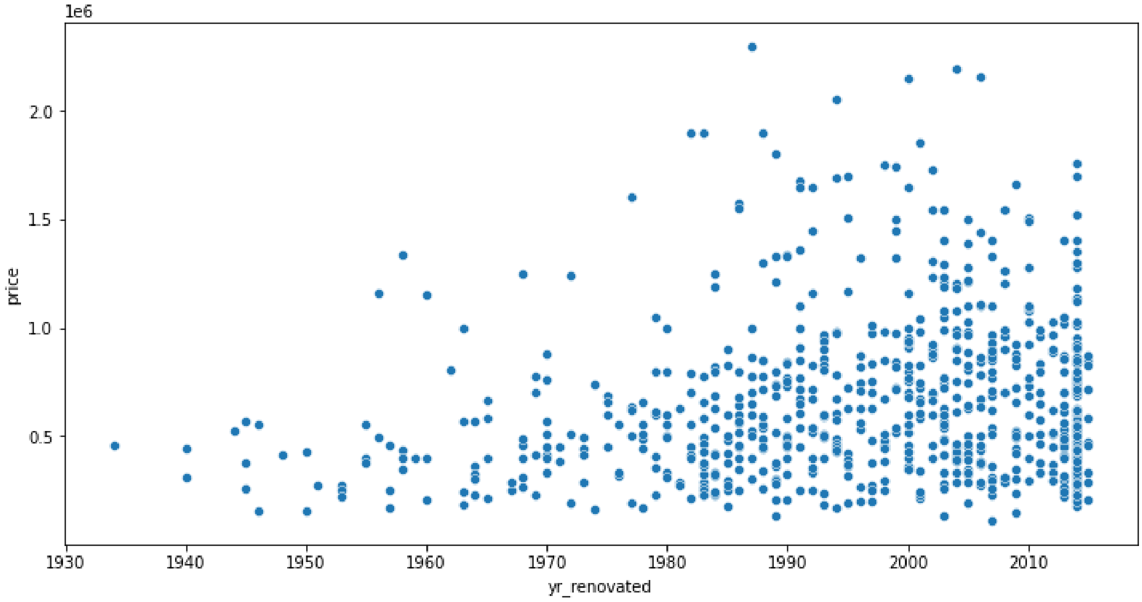
yr_renovated

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

Out[102]:

	mean	median	size
yr_renovated			
1934	459950.000000	459950.0	1
1940	378400.000000	378400.0	2
1944	521000.000000	521000.0	1
1945	398666.666667	375000.0	3
1946	351137.500000	351137.5	2
...	...	...	...
2011	607496.153846	577000.0	13
2012	625181.818182	515000.0	11
2013	600985.000000	518500.0	30

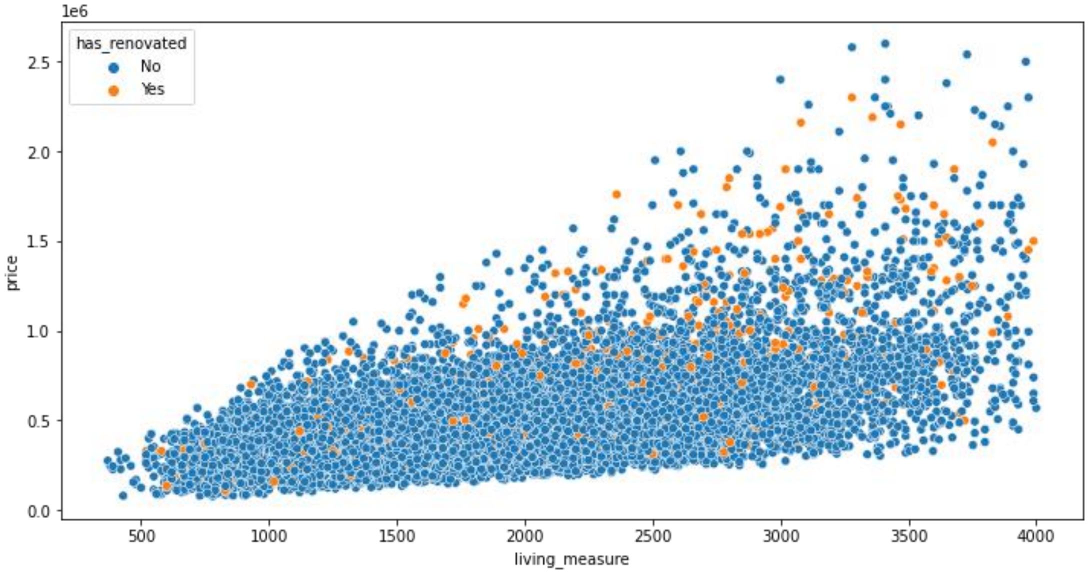
	mean	median	size
yr_renovated			
2014	655652.250000	587000.0	84
2015	561250.000000	530500.0	10



So most houses are renovated after 1980's. We will create new categorical variable 'has_renovated' to categorize the property as renovated and non-renovated. For further ananlysis we will use this categorical variable.

Creating new categorical column for looking that a house is renovated or not

	price			house_land_ratio		
	mean	median	size	mean	median	size
has_renovated						
No	478764.60231	425000.0	17574	23.907249	21.0	17574
Yes	665100.99021	575000.0	715	25.060140	24.0	715

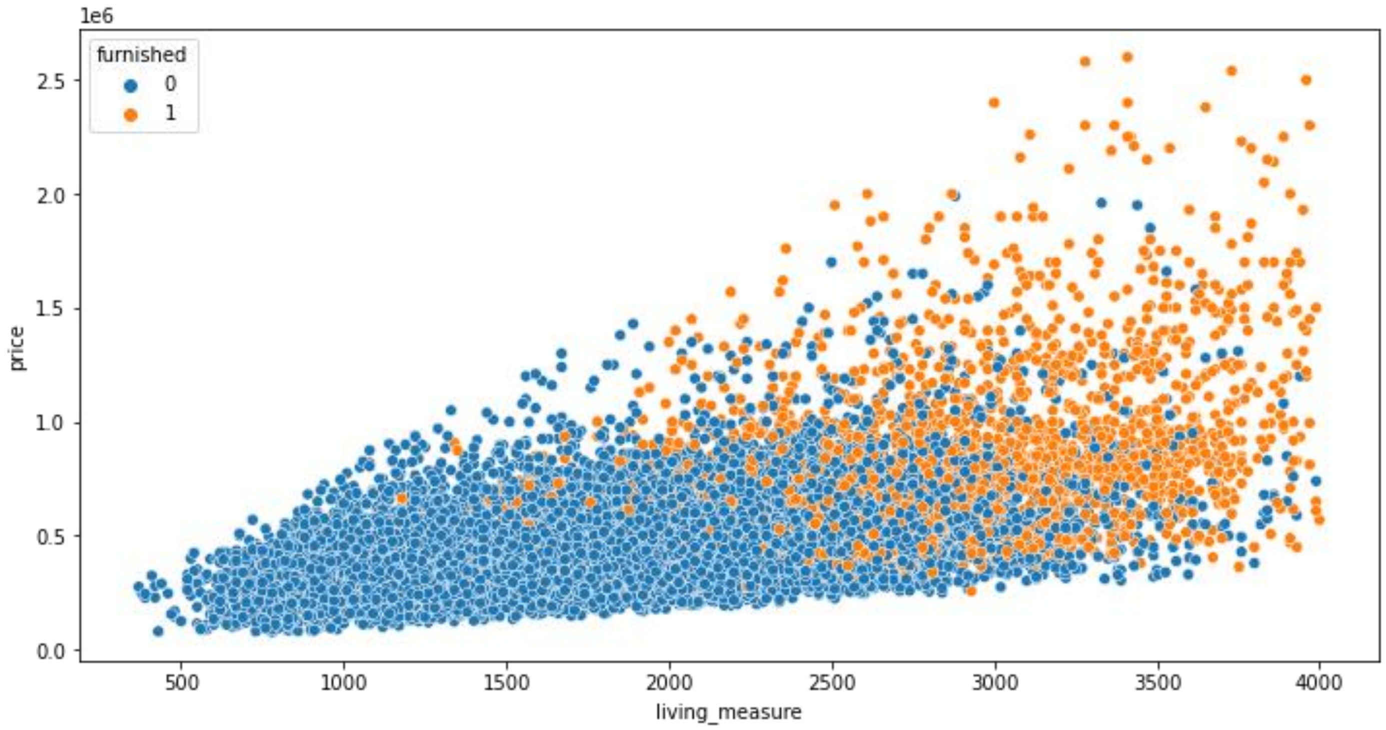


Renovated house utilized more land area for construction of house

Renovated properties have higher price than others with same living measure space.

Furnished

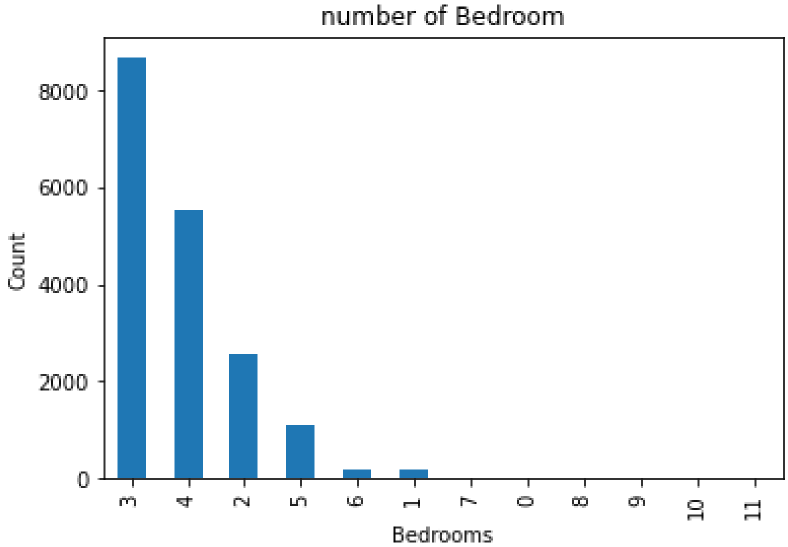
	price			living_measure			house_land_ratio		
	mean	median	size	mean	median	size	mean	median	size
furnished									
0	430609.413321	398000.0	15690	1745.599363	1690.0	15690	23.008540	20.0	15690
1	820736.681031	755000.0	2599	2829.731820	2850.0	2599	29.649865	29.0	2599



Furnished has higher price value and has greater living_measure. Furnished houses have higher price than that of the Non-furnished houses

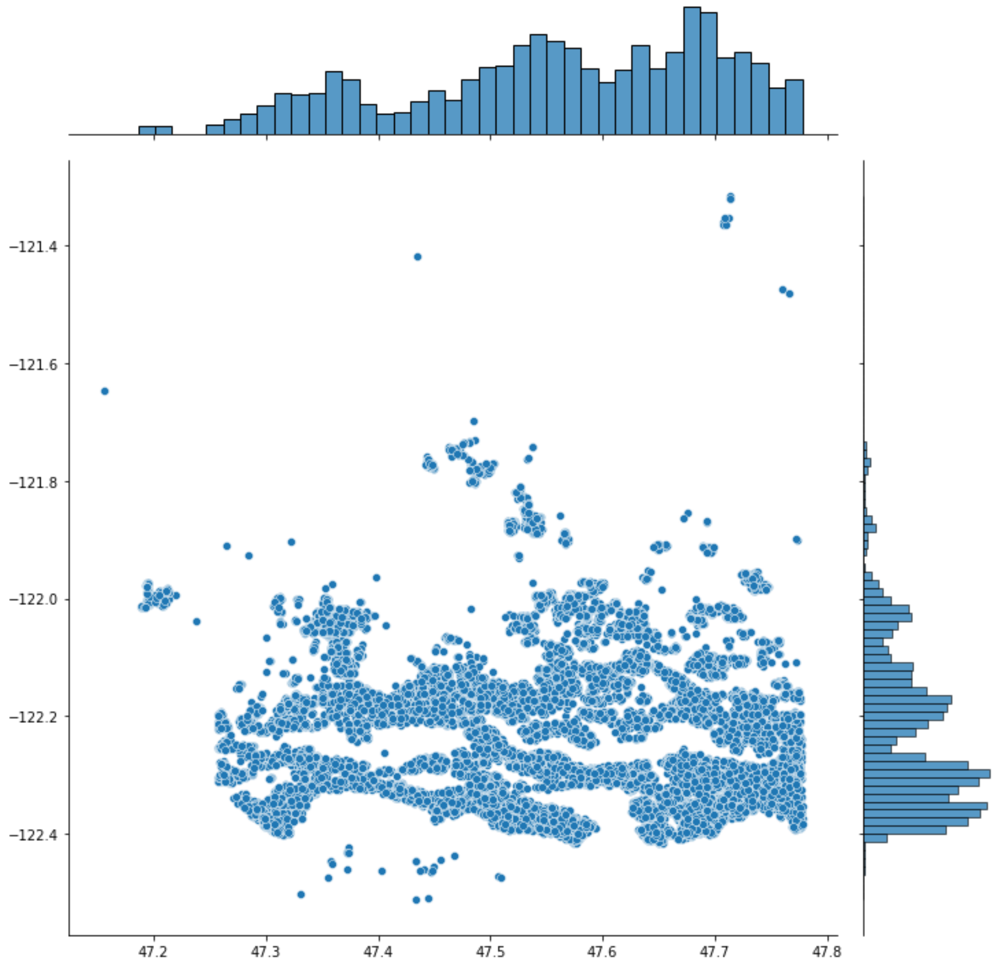
Some other analysis

Looking into the most common house according to number of bedroom



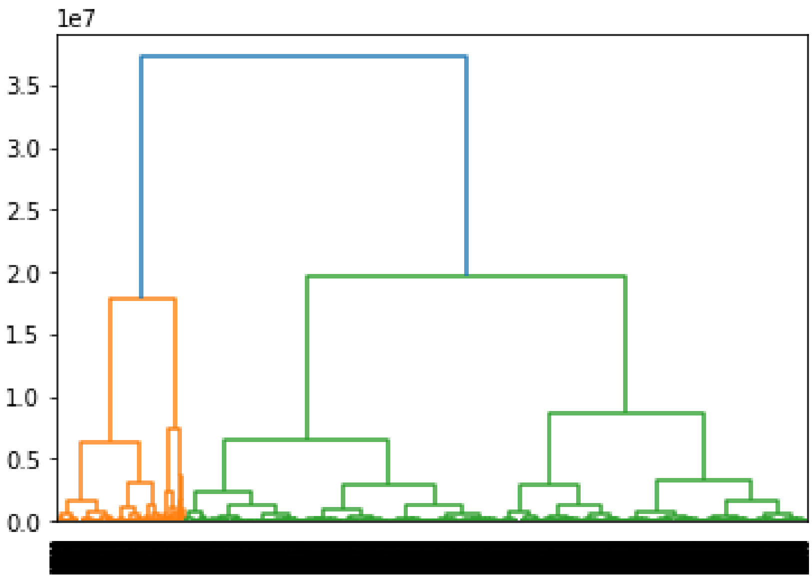
we can clearly see that the houses having 3 and 4 number of bedrooms are higher.

Visualizing the location of the houses based on latitude and longitude.



We can see that for latitude between -47.5 and -47.8 and for longitude between -122.0 to -122.4 there are many houses.

Hierarchical clustering & KMeans Clustering :



Value Counts:

```
1    3108
2    7718
3    7463
```

Aggregate mean data:

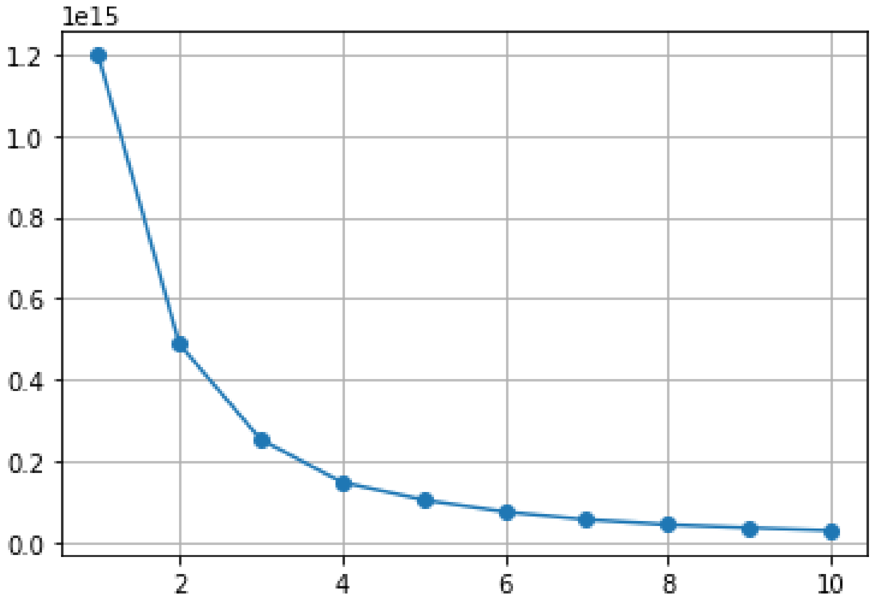
H_Clusters	price	living_measure	lot_measure	ceil_measure	basement	living_measure15	lot_measure15	total_area	house_land_ratio
1	917176.802445	2659.384170	7550.832368	2259.844916	399.833655	2432.795689	7495.251287	10214.236808	28.746139
2	286988.579166	1560.339077	7482.263799	1394.186447	166.026950	1615.950635	7565.578259	9046.197331	19.845038
3	512366.738979	1934.190540	6645.909822	1645.874581	288.145786	1888.342490	6682.221359	8579.213721	26.203537

Conclusion:

Here, we can clearly see that hierarchical cluster method shows us 3 group of clusters. The 1st Group belongs to the highest price range group the 3rd one belongs to the middle price range group & 2nd belongs to the lowest price range group. The group 1 posses 3108 properties, group 2 with 7718 properties & group 3 with 7463 properties. As per this we can come to a decision that most of the properties falls in lowest price group then the middle one followed by highest price group.

KMeans Clustering:

Elbow Chart:



Value Counts:

```
0     6268
1    11275
2      746
```

Aggregate mean data:

KMeans_Clus	price	living_measure	lot_measure	ceil_measure	basement	living_measure15	lot_measure15	total_area	house_land_ratio
0	6.601877e+05	2251.608966	6917.754308	1916.193523	335.484046	2132.674059	6905.249362	9168.482451	27.653797
1	3.351264e+05	1632.379069	7225.096497	1439.219690	193.003725	1667.743503	7291.222350	8859.542971	21.483636
2	1.303954e+06	2982.262735	8030.934316	2451.994638	530.268097	2619.741287	8130.237265	11034.418231	30.163539

Conclusion:

Here, we can see that 3 groups are formed first is group 0 with 6268 propertise then group 1 with 11275 propertise followed by group 2 with 746 propertise from this we can see that most of the propertise falls in group 1 followed by group 0 & group 2. Group 1 have a price range which comes in between group 0 & 2 whereas group 0 have maximum price range & group 2 have the lowest price range.So, it will be good to invest in those properties which falls in group 1.

INSIGHTS:

So, from the data analysis we can see that data is somewhat unbalanced so we have to balance that also we can see that houses with basement have higher prices but those houses are very low in number so we should build house with basement in them also the sights have not been seen so much we can have an advertisement ready for that matter as much people can get aware that house is on sale and also we have seen that quality of most of the houses are graded are at 3 so we can work on the quality of the houses so that we house can also be sold easily and also the prices can be

increased. We can also see that people are more interested in buying houses with more bedrooms so we should build houses with more bedrooms and also bathrooms people also like to buy houses with more bathrooms and we can see that people are tend to buy house which have 2 floors in it so we should mainly concentrate on building 2 floor houses. There are not many houses furnished so we should pay attention as people are more likely to buy furnished houses now days. As for other variables I would like to suggest we can apply a strategy to inform people about the things they are not aware off because right now people don’t pay attention on their houses we have to make them aware and guide so that we can also get business out of them we can also create a app and can make people aware about it so that we can get more data and information about their way of living and also their behavior according to which we can take action and also make business out of it.

Checking P-Values & Co-efficients of the variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.576
Model:                            OLS   Adj. R-squared:                  0.574
Method:                 Least Squares   F-statistic:                     368.1
Date:                Wed, 08 Feb 2023   Prob (F-statistic):               0.00
Time:                        22:01:47   Log-Likelihood:            -1.7222e+05
No. Observations:               12802   AIC:                         3.445e+05
Df Residuals:                   12754   BIC:                         3.449e+05
Df Model:                          47
Covariance Type:            nonrobust
==================================================================================
==
                        coef    std err          t      P>|t|      [0.025      0.97
5]
----------------------------------------------------------------------------------
--
const              2.487e+05   1.06e+05      2.354      0.019    4.16e+04    4.56e+
05
living_measure       81.3005    138.553      0.587      0.557    -190.284      352.8
85
lot_measure         -19.3871      9.160     -2.117      0.034     -37.342       -1.4
32
ceil_measure          5.8276    138.767      0.042      0.967    -266.177      277.8
32
basement            -26.3199    138.971     -0.189      0.850    -298.723      246.0
83
living_measure15     65.1966      4.325     15.074      0.000      56.719       73.6
74
```

lot_measure15	-3.1378	0.532	-5.894	0.000	-4.181	-2.094
total_area	15.3173	9.105	1.682	0.093	-2.529	33.164
house_land_ratio	1629.2380	353.800	4.605	0.000	935.736	2322.740
room_bed_1	7.772e+04	7.9e+04	0.984	0.325	-7.71e+04	2.33e+05
room_bed_2	7.19e+04	7.77e+04	0.925	0.355	-8.04e+04	2.24e+05
room_bed_3	4.098e+04	7.76e+04	0.528	0.598	-1.11e+05	1.93e+05
room_bed_4	2.889e+04	7.77e+04	0.372	0.710	-1.23e+05	1.81e+05
room_bed_5	3.976e+04	7.79e+04	0.510	0.610	-1.13e+05	1.93e+05
room_bed_6	2.47e+04	7.91e+04	0.312	0.755	-1.3e+05	1.8e+05
room_bed_7	-3.006e+04	9e+04	-0.334	0.738	-2.06e+05	1.46e+05
room_bed_8	3.467e+04	1.01e+05	0.343	0.732	-1.64e+05	2.33e+05
room_bed_9	3.03e+05	1.25e+05	2.427	0.015	5.83e+04	5.48e+05
room_bed_10	1.172e+05	1.42e+05	0.822	0.411	-1.62e+05	3.96e+05
room_bed_11	-7.586e+04	1.86e+05	-0.408	0.683	-4.4e+05	2.89e+05
room_bath_1	1.167e+05	9.78e+04	1.193	0.233	-7.51e+04	3.09e+05
room_bath_2	7.522e+04	9.78e+04	0.769	0.442	-1.17e+05	2.67e+05
room_bath_3	8.507e+04	9.8e+04	0.868	0.385	-1.07e+05	2.77e+05
room_bath_4	1.362e+05	9.83e+04	1.386	0.166	-5.65e+04	3.29e+05
room_bath_5	3.81e+05	1.18e+05	3.233	0.001	1.5e+05	6.12e+05
room_bath_6	-2.182e+05	2.01e+05	-1.087	0.277	-6.12e+05	1.75e+05
ceil_2	-5.684e+04	5050.953	-11.253	0.000	-6.67e+04	-4.69e+04
ceil_3	-1.669e+04	1.17e+04	-1.424	0.155	-3.97e+04	6289.085
coast_1	3.081e+05	3.12e+04	9.880	0.000	2.47e+05	3.69e+05
sight_1	1.114e+05	1.24e+04	9.015	0.000	8.72e+04	1.36e+05
sight_2	7.99e+04	8051.694	9.923	0.000	6.41e+04	9.57e+04
sight_3	1.358e+05	1.18e+04	11.490	0.000	1.13e+05	1.59e+05
sight_4	2.829e+05	1.99e+04	14.187	0.000	2.44e+05	3.22e+05
condition_2	2.687e+04	4.55e+04	0.590	0.555	-6.24e+04	1.16e+05
condition_3	1.877e+04	4.24e+04	0.443	0.658	-6.44e+04	1.02e+05
condition_4	8.277e+04	4.25e+04	1.949	0.051	-469.800	1.66e+05
condition_5	1.384e+05	4.27e+04	3.243	0.001	5.48e+04	2.22e+05

```
quality_4          -4.023e+05   7.95e+04     -5.062      0.000    -5.58e+05   -2.47e+
05
quality_5          -4.028e+05   7.06e+04     -5.706      0.000    -5.41e+05   -2.64e+
05
quality_6          -3.681e+05   6.96e+04     -5.292      0.000    -5.04e+05   -2.32e+
05
quality_7          -3.001e+05   6.94e+04     -4.322      0.000    -4.36e+05   -1.64e+
05
quality_8          -2.174e+05   6.95e+04     -3.130      0.002    -3.54e+05   -8.12e+
04
quality_9          -4559.2433   1.03e+05     -0.044      0.965    -2.06e+05    1.97e+
05
quality_10           1.45e+05   1.03e+05      1.408      0.159    -5.68e+04    3.47e+
05
quality_11           3.64e+05   1.05e+05      3.478      0.001     1.59e+05    5.69e+
05
quality_12          1.435e+06   1.48e+05      9.679      0.000     1.14e+06    1.73e+
06
furnished_1        -6.898e+04   1.69e+05     -0.408      0.683       -4e+05    2.62e+
05
has_basement_Yes    5.173e+04   6465.506      8.000      0.000     3.91e+04    6.44e+
04
has_renovated_Yes   1.418e+05   7930.793     17.876      0.000     1.26e+05    1.57e+
05
==============================================================================
Omnibus:                     3779.087   Durbin-Watson:                   1.987
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            22337.381
Skew:                           1.287   Prob(JB):                         0.00
Kurtosis:                       8.937   Cond. No.                     2.72e+15
==============================================================================
```

Here, we can look at the co-efficients we can interpret values based on if it's positive and negative. whichever variables co-efficient is positive meaning for e.g. every increase in value of living_measure there is price increase we can apply the same logic in every variable which is positive and whichever variable co-efficient is negative for eg if there are negative co-efficients variable is tend to lose the price or can say there is decrease in price according to the value that the variable has. For, P - Values we can say whichever variable has a higher value then 0.05 that is insignificant for us and the variables which are under or equal to 0.05 are significant values can say important variables for prediction on our data.

TRAIN FINAL

	Method	Train	RMSE	MSE	MAE
0	RF_Train	0.947836	59006.801046	3.481803e+09	41223.896950
0	BGG_Train	0.926951	69827.038668	4.875815e+09	45839.162591
0	SVR_Train	0.998463	10129.164104	1.026000e+08	180137.556866
0	NB_Train	0.999086	7811.557350	6.102043e+07	137477.887049
0	KNN_Train	0.999987	920.225588	8.468151e+05	972.428995
0	LogisticReg_Train	0.998886	8624.409894	7.438045e+07	217180.534995
0	LinearReg_Model_Train	0.575639	168300.179407	2.832495e+10	121310.244780
0	LDA_model_Train	0.998662	9449.488652	8.929284e+07	138367.520856
0	DT_Train	0.998478	10078.403407	1.015742e+08	972.428995
0	GB_Train	0.666810	149129.118645	2.223949e+10	109406.273650

Test Final

	Method	Test	RMSE	MSE	MAE
0	RF_Test	0.635873	150996.330276	2.279989e+10	108319.873557
0	BGG_Test	0.597312	158790.364726	2.521438e+10	113494.111201
0	SVR_Test	0.998643	9218.409793	8.497908e+07	178079.348278
0	NB_Test	0.998267	10417.914480	1.085329e+08	165337.612903
0	KNN_Test	0.998556	9509.663910	9.043371e+07	168178.955349
0	LogisticReg_Test	0.999130	7379.569804	5.445805e+07	214311.377438

	Method	Test	RMSE	MSE	MAE
0	LinearReg_Model_Test	0.560861	165821.547377	2.749679e+10	120719.887927
0	LDA_model_Test	0.998278	10383.348332	1.078139e+08	154302.666484
0	DT_Test	0.291741	210589.222914	4.434782e+10	146239.745125
0	GB_Test	0.620841	154081.610408	2.374114e+10	112110.008184

Here, we have 10 models in which we can see both RMSE value and Train and Test value either we can go with Train and Test values or we can go with RMSE values i prefer to go with Train and Test value so fo that reason the most suited model would be the RF Model because other models are either overfitted or have low value and if see from the perspective of RMSE value If the noise is small, as estimated by RMSE, this generally means our model is good at predicting our observed data, and if RMSE is large, this generally means our model is failing to account for important features underlying our data.So, These models will help us anlyse the most appropriate way to what direction we should go it will help us determine correct prices and areas and all of the variables which are present in our data.

After applying Hyperparameters

Train

	Method	Train	RMSE	MSE	MAE
0	LinearReg_Model_Train	0.251387	223534.894992	4.996785e+10	157254.849946
0	BGG_Train	0.928337	69161.497089	4.783313e+09	47559.923003
0	RF_Train	0.999978	1208.240889	1.459846e+06	1079.713404
0	NB_Train	0.999683	4597.865139	2.114036e+07	406970.755898

	Method	Train	RMSE	MSE	MAE
0	KNN_Train	0.999987	920.225588	8.468151e+05	972.428995
0	LDA_model_Train	0.998732	9201.096740	8.466018e+07	139243.645758
0	DT_Train	0.998942	8401.502000	7.058524e+07	145354.971176
0	GB_Train	0.585927	166247.551818	2.763825e+10	121298.552430

Test

	Method	Test	RMSE	MSE	MAE
0	LinearReg_Model_Test	0.238240	218398.198028	4.769777e+10	156968.687434
0	BGG_Test	0.585227	161155.557729	2.597111e+10	115886.435250
0	RF_Test	0.997833	11647.674750	1.356683e+08	142296.090760
0	NB_Test	0.999849	3079.374136	9.482545e+06	403900.257336
0	KNN_Test	0.998523	9617.524764	9.249678e+07	168105.524330
0	LDA_model_Test	0.998385	10057.515745	1.011536e+08	157340.896847
0	DT_Test	0.998288	10354.402630	1.072137e+08	153564.095134
0	GB_Test	0.566994	164659.570140	2.711277e+10	121791.912954

Here, we can see that some of the models are not been ran becasue my computer ran out of memory and wasnr able to cope up with the processimg luckily i have a table in my pdf the only model it doesnt consist is logistic regression and SVR. so, here we can see that most of the model are overfitted so here i would choose BGG Model as its not overfitted and can provide good assistance while applying this model so yeah in all i would prefer the models in which i have not used hyperparameters that is RF model its the most optimum model and better suited for the business and the direction it want to go in.

Insights & Recommendation:

- **Here, we can see that properties which are falling in one of the 3 categories in both the methods which we used to make clusters from observing that we can predict the price of those properties based on that result also we can probably take a hunch of the price of the property if its going to increase in the near future based on this current data.**
- **Here, I have used various models for predictions. So, that we can get accuracy on our price prediction through the Train & Test values of our models , we can also choose one model if we have a preference to choose by RMSE values.**
- **We also have values of MAE & MSE which helps us decide which model to choose. Mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon & the Mean squared error (MSE) of an estimator measures the average of the squares of the errors that is, the average squared difference between the estimated values and the actual value.**
- **I prefer to go with Train and Test value so for that reason the most suited model would be the RF Model because other models are either over fitted or have low value and if see from the perspective of RMSE value If the noise is small, as estimated by RMSE, this generally means our model is good at predicting our observed data, and if RMSE is large, this generally means our model is failing to account for important features underlying our data. So, These models will help us analyze the most appropriate way to what direction we should go it will help us determine correct prices and areas and all of the variables which are present in our data.**
- **so, here in 2ns Section of Table which we got by Tuning our models. We can see that most of the model are over fitted. So, here I would choose BGG Model as its not over fitted and can**

provide good assistance while applying this model. So, yeah in all I would prefer the models in which I have not used hyper parameters that is RF model its the most optimum model and better suited for the business and the direction it want to go in.

- **So, I would suggest you if you want to sell your house please keep in mind the Value which we generated from cluster. I'm going to choose the price around 350,000 $ which is an average price but if your home is in a good location and have more no of bedrooms & bathroom and is fully furnished and have a coastal view you can charge around 550,000$ - 600,000$.**