

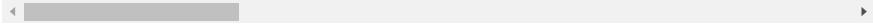
```
# importing required libraries
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# load the dataset
df = pd.read_csv('/content/House Price India.csv')
df.head()
```

	id	Date	number_of_bedrooms	number_of_bathrooms	living_area	lot_area	
0	6762810145	42491		5	2.50	3650	9050
1	6762810635	42491		4	2.50	2920	4000
2	6762810998	42491		5	2.75	2910	9480
3	6762812605	42491		4	2.50	3310	42998
4	6762812919	42491		3	2.00	2710	4500

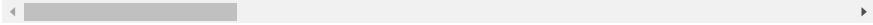
5 rows × 23 columns



```
df.tail()
```

	id	Date	number_of_bedrooms	number_of_bathrooms	living_area	lot_area	
14615	6762830250	42734		2	1.5	1556	2000
14616	6762830339	42734		3	2.0	1680	2000
14617	6762830618	42734		2	1.0	1070	1000
14618	6762830709	42734		4	1.0	1030	1000
14619	6762831463	42734		3	1.0	900	1000

5 rows × 23 columns



```
df.shape
```

(14620, 23)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               14620 non-null   int64  
 1   Date              14620 non-null   int64  
 2   number_of_bedrooms 14620 non-null   int64  
 3   number_of_bathrooms 14620 non-null   float64 
 4   living_area        14620 non-null   int64  
 5   lot_area           14620 non-null   int64  
 6   number_of_floors    14620 non-null   float64 
 7   waterfront_present 14620 non-null   int64  
 8   number_of_views     14620 non-null   int64  
 9   condition_of_the_house 14620 non-null   int64  
 10  grade_of_the_house  14620 non-null   int64  
 11  Area_of_the_house(excluding basement) 14620 non-null   int64  
 12  Area_of_the_basement 14620 non-null   int64  
 13  Built_Year         14620 non-null   int64  
 14  Renovation_Year    14620 non-null   int64  
 15  Postal_Code        14620 non-null   int64  
 16  Latitude            14620 non-null   float64 
 17  Longitude           14620 non-null   float64 
 18  living_area_renov   14620 non-null   int64  
 19  lot_area_renov      14620 non-null   int64  
 20  Number_of_schools_nearby 14620 non-null   int64  
 21  Distance_from_the_airport 14620 non-null   int64  
 22  Price               14620 non-null   int64  
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

```
df.isnull().any()
```

```

id                         False
Date                        False
number_of_bedrooms          False
number_of_bathrooms          False
living_area                  False
lot_area                     False
number_of_floors             False
waterfront_present           False
number_of_views               False
condition_of_the_house      False
grade_of_the_house           False
Area_of_the_house(excluding basement) False
Area_of_the_basement        False
Built_Year                   False
Renovation_Year              False
Postal_Code                 False
Latitude                    False
Longitude                   False
living_area_renov            False
lot_area_renov               False
Number_of_schools_nearby    False
Distance_from_the_airport   False
Price                        False
dtype: bool

```

```
df.describe()
```

	id	Date	number_of_bedrooms	number_of_bathrooms	living_ar
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.2625
std	6.237575e+03	67.347991	0.938719	0.769934	928.2757
min	6.762810e+09	42491.000000	1.000000	0.500000	370.0000
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.0000
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.0000
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.0000
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.0000

8 rows × 23 columns

```
df.Date.unique()
```

```

array([42491, 42492, 42493, 42494, 42495, 42496, 42497, 42498, 42499,
       42500, 42501, 42502, 42503, 42504, 42505, 42506, 42507, 42508,
       42509, 42510, 42511, 42512, 42513, 42514, 42515, 42516, 42517,
       42518, 42519, 42520, 42521, 42522, 42523, 42524, 42525, 42526,
       42527, 42528, 42529, 42530, 42531, 42532, 42533, 42534, 42535,
       42536, 42537, 42538, 42539, 42540, 42541, 42542, 42543, 42544,
       42545, 42546, 42547, 42548, 42549, 42550, 42551, 42552, 42553,
       42554, 42555, 42556, 42557, 42558, 42559, 42560, 42561, 42562,
       42563, 42564, 42565, 42566, 42567, 42568, 42569, 42570, 42571,
       42572, 42573, 42574, 42575, 42576, 42577, 42578, 42579, 42580,
       42581, 42582, 42583, 42584, 42585, 42586, 42587, 42588, 42589,
       42590, 42591, 42592, 42593, 42594, 42595, 42596, 42597, 42598,
       42599, 42600, 42601, 42602, 42603, 42604, 42605, 42606, 42607,
       42608, 42609, 42610, 42611, 42612, 42613, 42614, 42615, 42616,
       42617, 42618, 42619, 42620, 42621, 42622, 42623, 42624, 42625,
       42626, 42627, 42628, 42629, 42630, 42631, 42632, 42633, 42634,
       42635, 42636, 42637, 42638, 42639, 42640, 42641, 42642, 42643,
       42644, 42645, 42646, 42647, 42648, 42649, 42650, 42651, 42652,
       42653, 42654, 42655, 42656, 42657, 42658, 42659, 42660, 42661,
       42662, 42663, 42664, 42665, 42666, 42667, 42668, 42669, 42670,
       42671, 42672, 42673, 42674, 42675, 42676, 42677, 42678, 42679,
       42680, 42681, 42682, 42683, 42684, 42685, 42686, 42687, 42688,
       42689, 42690, 42691, 42692, 42693, 42694, 42695, 42696, 42697,
       42698, 42699, 42701, 42702, 42703, 42704, 42705, 42706, 42707,
       42708, 42709, 42710, 42711, 42712, 42713, 42714, 42715, 42716,
       42717, 42718, 42719, 42720, 42721, 42722, 42723, 42724, 42725,
       42726, 42727, 42729, 42730, 42732, 42733, 42734])

```

```
df.number_of_bedrooms.value_counts()
```

3	6612
4	4724
2	1844
5	1079
6	176
1	136
7	30

```

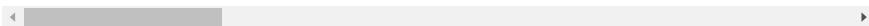
8      11
9      3
10     3
33     1
11     1
Name: number_of_bedrooms, dtype: int64

```

```
df.corr()
```

	id	Date	number_of_bedrooms	number_of_bathro
id	1.000000	0.045966	-0.329034	-0.516
Date	0.045966	1.000000	-0.015663	-0.026
number_of_bedrooms	-0.329034	-0.015663	1.000000	0.509
number_of_bathrooms	-0.516909	-0.026485	0.509784	1.000
living_area	-0.648127	-0.021958	0.570526	0.753
lot_area	-0.100269	0.004392	0.034416	0.080
number_of_floors	-0.312305	-0.010335	0.177294	0.502
waterfront_present	-0.112937	0.012006	-0.006257	0.060
number_of_views	-0.293004	-0.004782	0.078665	0.183
condition_of_the_house	-0.045061	-0.027402	0.026597	-0.128
grade_of_the_house	-0.673448	-0.033097	0.352945	0.663
Area_of_the_house(excluding basement)	-0.565116	-0.015994	0.473599	0.684
Area_of_the_basement	-0.290806	-0.015711	0.300332	0.287
Built_Year	-0.068645	-0.005869	0.152954	0.498
Renovation_Year	-0.109155	-0.011636	0.016132	0.049
Postal_Code	0.294709	0.018243	-0.044156	-0.105
Latitude	-0.479334	-0.023327	-0.013163	0.031
Longitude	-0.070841	-0.018231	0.135712	0.223
living_area_renov	-0.599900	-0.032495	0.389855	0.570
lot_area_renov	-0.089604	-0.000050	0.029400	0.078
Number_of_schools_nearby	-0.004821	-0.004071	0.003397	0.002
Distance_from_the_airport	-0.004542	0.011457	-0.006157	0.009
Price	-0.773114	-0.027919	0.308460	0.531

23 rows × 23 columns



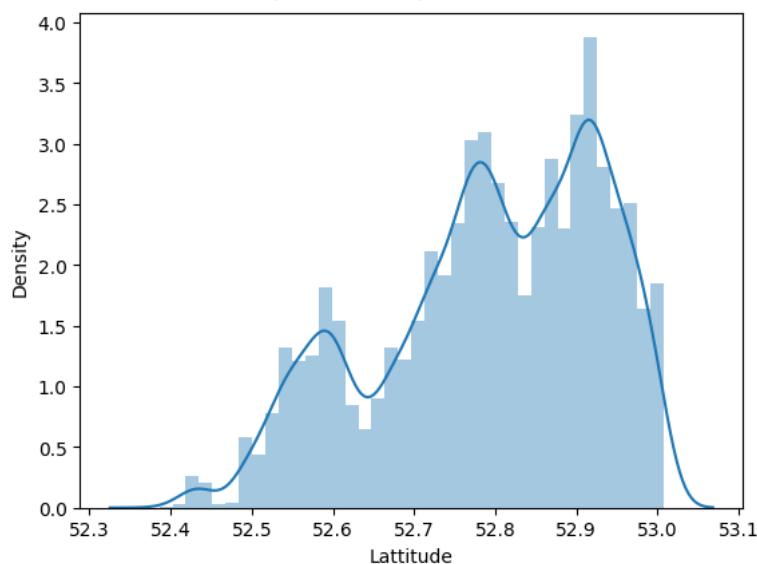
```
sns.distplot(df.number_of_bedrooms)
```

```
<ipython-input-22-b25ff7c8a519>:1: UserWarning:  
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.  
Please adapt your code to use either `displot` (a figure-level function with  
similar flexibility) or `histplot` (an axes-level function for histograms).  
For a guide to updating your code to use the new functions, please see  
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(df.Latitude)
```

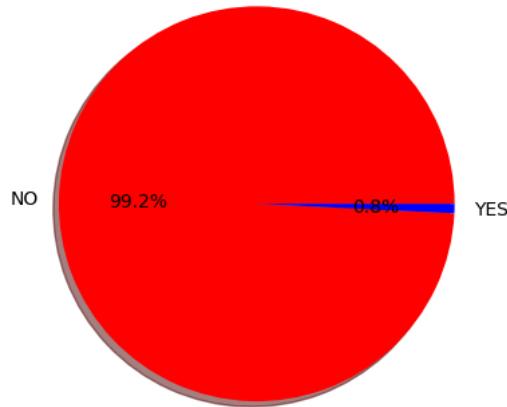
```
<ipython-input-23-4790d4fed0b1>:1: UserWarning:  
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.  
Please adapt your code to use either `displot` (a figure-level function with  
similar flexibility) or `histplot` (an axes-level function for histograms).  
For a guide to updating your code to use the new functions, please see  
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(df.Latitude)  
<Axes: xlabel='Latitude', ylabel='Density'>
```



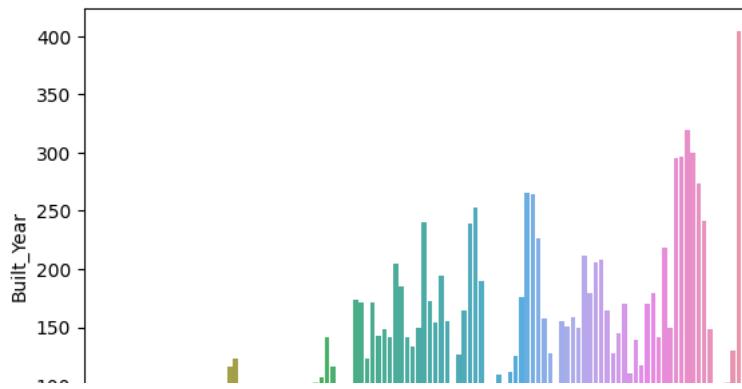
```
plt.pie(df.waterfront_present.value_counts(), labels=['NO', 'YES'], autopct="%1.1f%%", colors=['red', 'blue'], shadow=True)  
plt.title('WATERFRONT')  
plt.show()
```

WATERFRONT



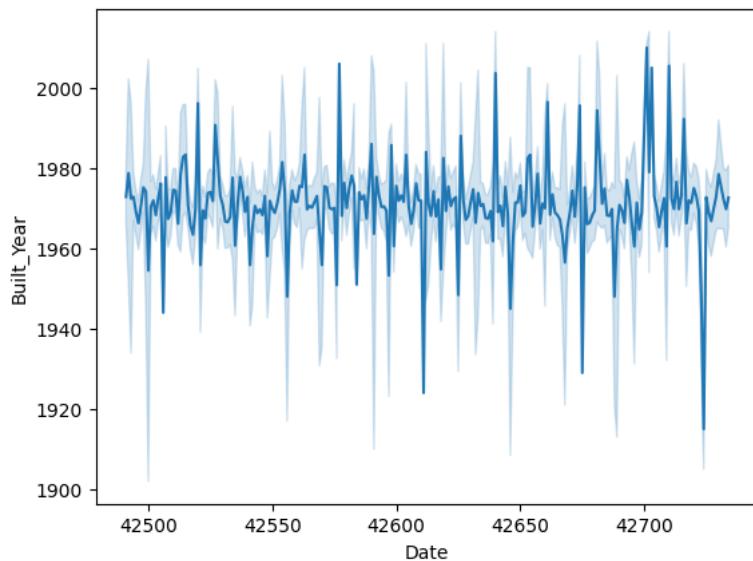
```
sns.barplot(x=df.Built_Year.value_counts().index,y=df.Built_Year.value_counts())
```

<Axes: ylabel='Built_Year'>



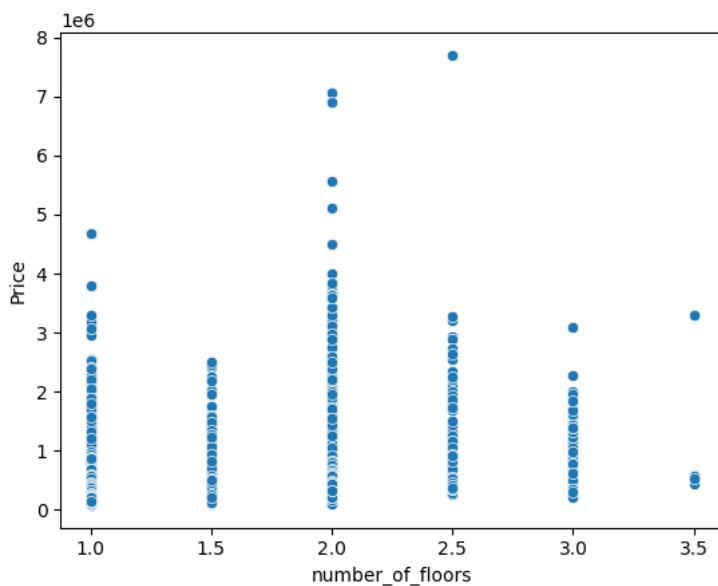
sns.lineplot(x=df.Date,y=df.Built_Year)

<Axes: xlabel='Date', ylabel='Built_Year'>



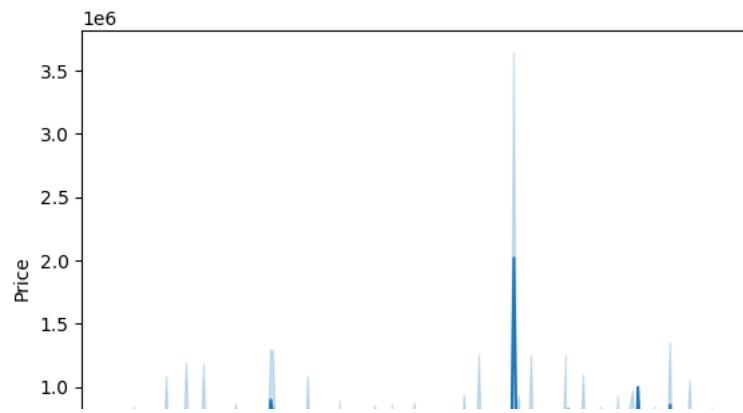
sns.scatterplot(x=df.number_of_floors,y=df.Price)

<Axes: xlabel='number_of_floors', ylabel='Price'>

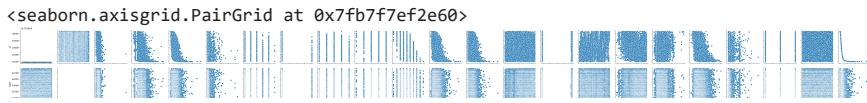


sns.lineplot(x=df.Date,y=df.Price)

<Axes: xlabel='Date', ylabel='Price'>



sns.pairplot(df)



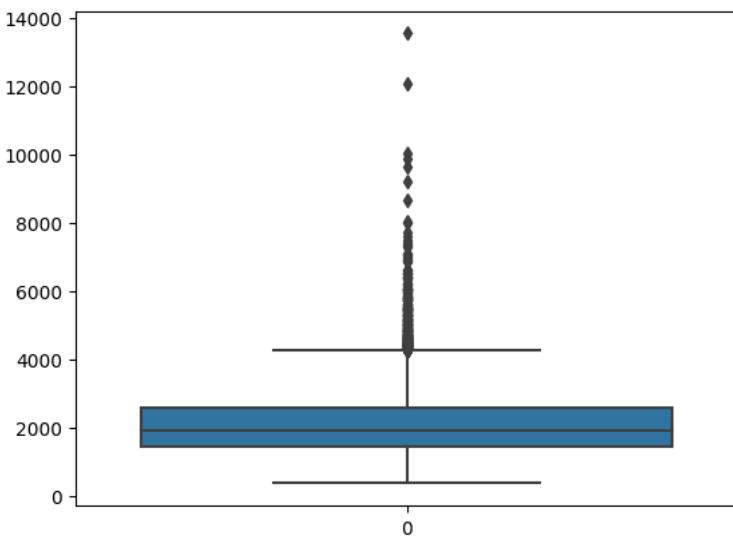
<Axes: >

id	-1.0
Date	-0.41
number_of_bedrooms	-0.311
number_of_bathrooms	-0.316
living_area	-0.652
lot_area	-0.104
number_of_floors	-0.118
waterfront_present	-0.101
number_of_views	-0.020
condition_of_the_house	-0.091
grade_of_the_house	-0.043
Area_of_the_house(excluding_basement)	-0.570
Area_of_the_basement	-0.291
Built_Year	-0.00559
Renovation_Year	-0.10120
Postal_Code	-0.990
Latitude	-0.482
Longitude	-0.700
living_area_renov	-0.05029
lot_area_renov	-0.56902
Number_of_schools_nearby	-0.04500
Distance_from_the_airport	-0.70281
Price	-0.70281



sns.boxplot(df.living_area)

<Axes: >



df.mean()

id	6.762821e+09
Date	4.260454e+04
number_of_bedrooms	3.379343e+00
number_of_bathrooms	2.129583e+00
living_area	2.098263e+03
lot_area	1.509328e+04
number_of_floors	1.502360e+00
waterfront_present	7.660739e-03
number_of_views	2.331053e-01

```

condition_of_the_house           3.430506e+00
grade_of_the_house              7.682421e+00
Area_of_the_house(excluding basement) 1.801784e+03
Area_of_the_basement            2.964791e+02
Built_Year                      1.970926e+03
Renovation_Year                 9.092401e+01
Postal_Code                      1.220331e+05
Latitude                         5.279285e+01
Longitude                        -1.144040e+02
living_area_renov                1.996702e+03
lot_area_renov                  1.275350e+04
Number_of_schools_nearby        2.012244e+00
Distance_from_the_airport       6.495096e+01
Price                            5.389322e+05
dtype: float64

```

```

df.mean(axis=1)

0      2.941470e+08
1      2.941041e+08
2      2.940958e+08
3      2.940832e+08
4      2.940783e+08
...
14615    2.940548e+08
14616    2.940537e+08
14617    2.940531e+08
14618    2.940530e+08
14619    2.940503e+08
Length: 14620, dtype: float64

```

```

df.median()

id                           6.762821e+09
Date                          4.260000e+04
number_of_bedrooms            3.000000e+00
number_of_bathrooms           2.250000e+00
living_area                   1.930000e+03
lot_area                      7.620000e+03
number_of_floors              1.500000e+00
waterfront_present            0.000000e+00
number_of_views                0.000000e+00
condition_of_the_house        3.000000e+00
grade_of_the_house             7.000000e+00
Area_of_the_house(excluding basement) 1.580000e+03
Area_of_the_basement          0.000000e+00
Built_Year                     1.975000e+03
Renovation_Year                0.000000e+00
Postal_Code                    1.220320e+05
Latitude                       5.280640e+01
Longitude                      -1.144210e+02
living_area_renov              1.850000e+03
lot_area_renov                 7.620000e+03
Number_of_schools_nearby      2.000000e+00
Distance_from_the_airport     6.500000e+01
Price                          4.500000e+05
dtype: float64

```

```

df.median(axis=1)

0      58.0000
1      52.8878
2      52.8852
3      52.9532
4      52.9047
...
14615    52.6191
14616    52.5075
14617    52.7289
14618    52.7157
14619    55.0000
Length: 14620, dtype: float64

```

```
df.mode()
```

	<code>id</code>	<code>Date</code>	<code>number_of_bedrooms</code>	<code>number_of_bathrooms</code>	<code>living_area</code>	<code>lot</code>
0	6762810020	42543.0		3.0	2.5	1400.0
1	6762810021	NaN		NaN	NaN	NaN
2	6762810022	NaN		NaN	NaN	NaN
3	6762810023	NaN		NaN	NaN	NaN
4	6762810026	NaN		NaN	NaN	NaN
...
14615	6762831611	NaN		NaN	NaN	NaN
14616	6762831612	NaN		NaN	NaN	NaN

```
df['Longitude'].std()
```

```
0.14132590314165383
```

```
df.skew()
```

<code>id</code>	-0.000802
<code>Date</code>	0.143747
<code>number_of_bedrooms</code>	2.663257
<code>number_of_bathrooms</code>	0.556663
<code>living_area</code>	1.538337
<code>lot_area</code>	10.155206
<code>number_of_floors</code>	0.586158
<code>waterfront_present</code>	11.294672
<code>number_of_views</code>	3.409219
<code>condition_of_the_house</code>	1.018018
<code>grade_of_the_house</code>	0.777584
<code>Area_of_the_house(excluding basement)</code>	1.436446
<code>Area_of_the_basement</code>	1.609744
<code>Built_Year</code>	-0.472049
<code>Renovation_Year</code>	4.359764
<code>Postal_Code</code>	0.227735
<code>Latitude</code>	-0.523831
<code>Longitude</code>	0.873803
<code>living_area_renov</code>	1.081959
<code>lot_area_renov</code>	7.774206
<code>Number_of_schools_nearby</code>	-0.022519
<code>Distance_from_the_airport</code>	0.006114
<code>Price</code>	4.269298

`dtype: float64`

```
df.kurt()
```

↳ <code>id</code>	-1.201221
<code>Date</code>	-1.130823
<code>number_of_bedrooms</code>	69.240310
<code>number_of_bathrooms</code>	1.588195
<code>living_area</code>	6.073617
<code>lot_area</code>	164.757273
<code>number_of_floors</code>	-0.523576
<code>waterfront_present</code>	125.586791
<code>number_of_views</code>	10.968839
<code>condition_of_the_house</code>	0.351359
<code>grade_of_the_house</code>	1.048022
<code>Area_of_the_house(excluding basement)</code>	3.402258
<code>Area_of_the_basement</code>	3.139635
<code>Built_Year</code>	-0.673474
<code>Renovation_Year</code>	17.011306
<code>Postal_Code</code>	-1.058364
<code>Latitude</code>	-0.619219
<code>Longitude</code>	0.950315
<code>living_area_renov</code>	1.428944
<code>lot_area_renov</code>	79.360403
<code>Number_of_schools_nearby</code>	-1.502552
<code>Distance_from_the_airport</code>	-1.203048
<code>Price</code>	40.321918

`dtype: float64`

✓ 0s completed at 11:10 PM

