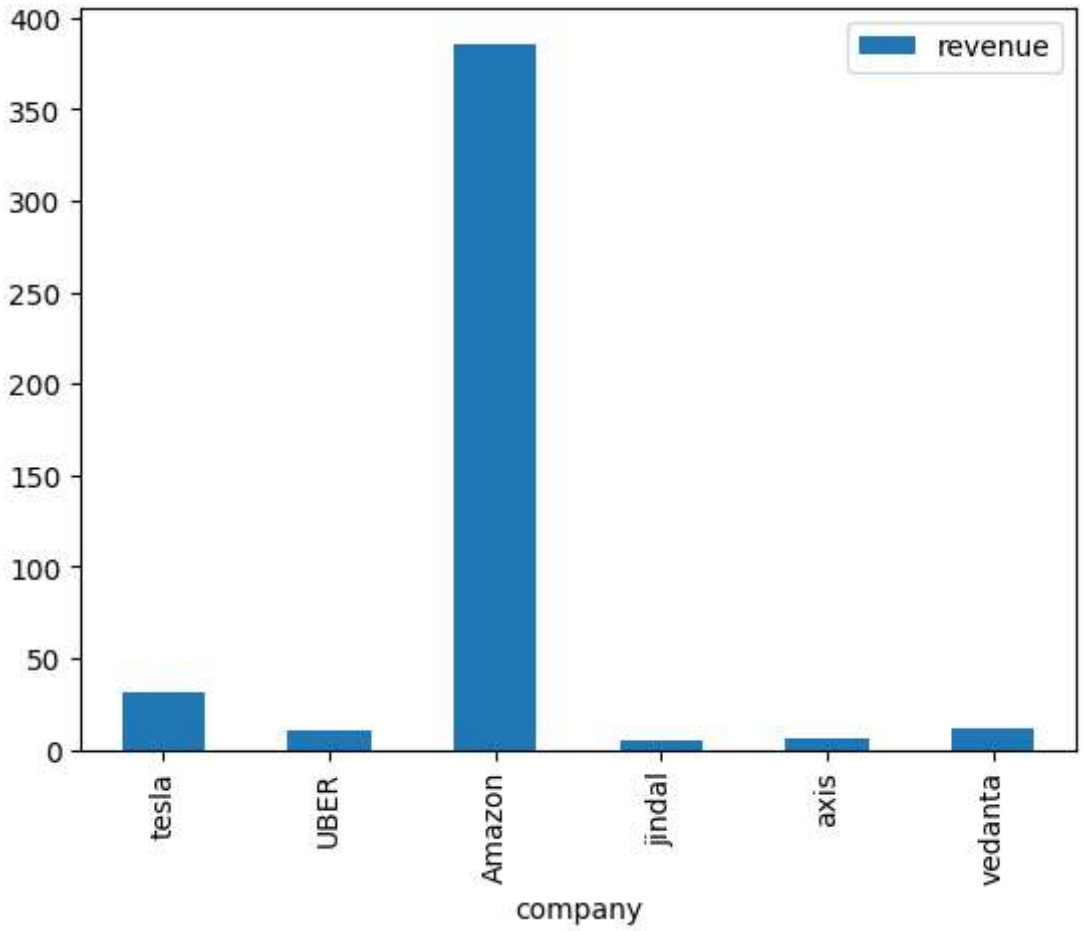
In [4]:
```
import pandas as pd
df = pd.read_csv('revenue.csv')
df.plot(x='company',y='revenue',kind='bar')
```

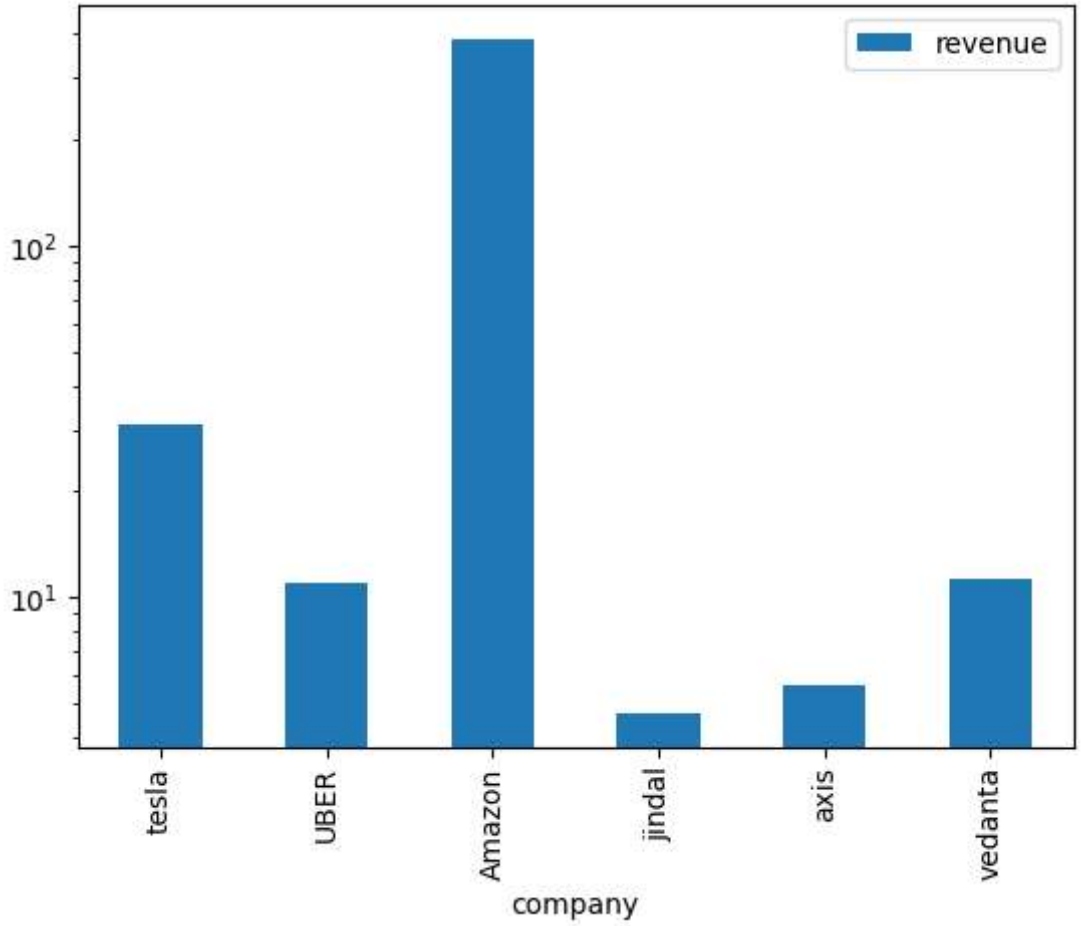
Out[4]: <AxesSubplot: xlabel='company'>



Chapter-4

In [6]:
```
df.plot(x='company',y='revenue',kind='bar',logy=True)
```

Out[6]: <AxesSubplot: xlabel='company'>



In [7]:
```
#histogram with a bell curve
import pandas as pd
import seaborn as sn
```

In [9]:
```
df = pd.read_csv('weight-height.csv')
df.head()
```

Out[9]:

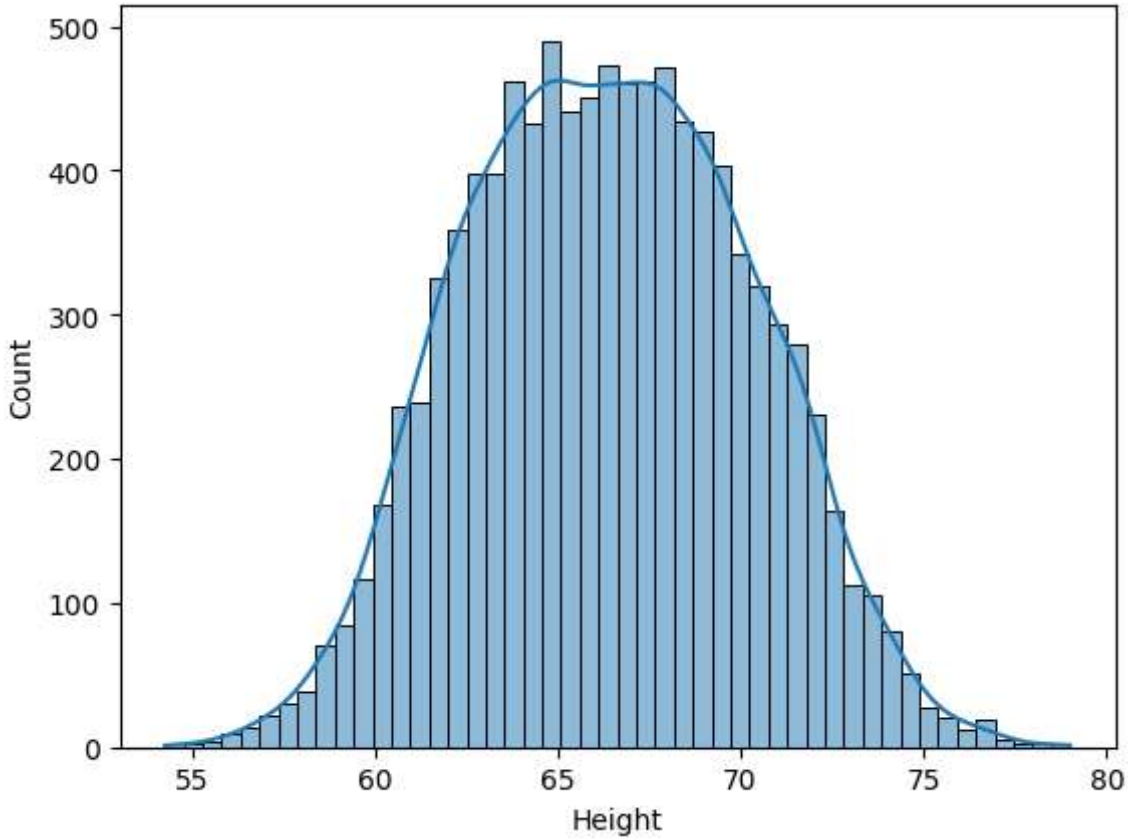
	Gender	Height
0	Male	73.847017
1	Male	68.781904
2	Male	74.110105
3	Male	71.730978
4	Male	69.881796

In [11]:
```
df.Height.describe()
```

Out[11]:
```
count    10000.000000
mean        66.367560
std          3.847528
min         54.263133
25%         63.505620
50%         66.318070
75%         69.174262
max         78.998742
Name: Height, dtype: float64
```

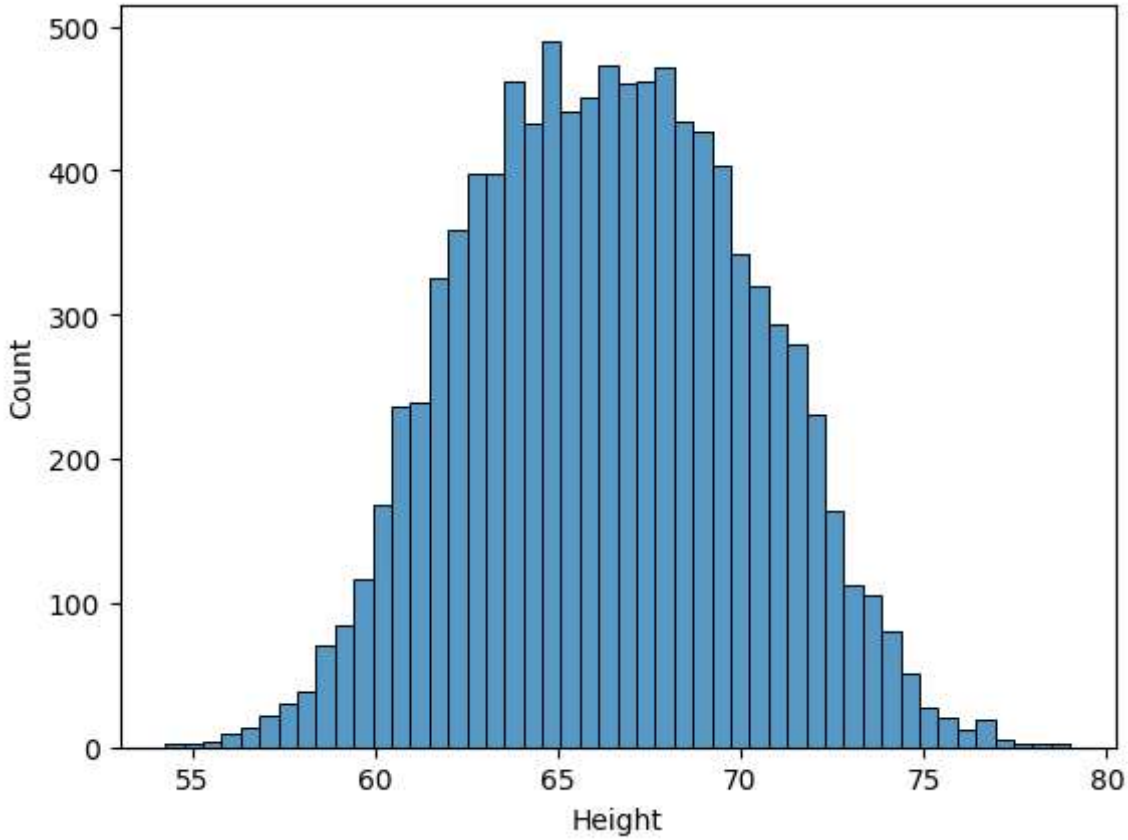
In [12]:
```
sn.histplot(df.Height, kde=True)
```

Out[12]:
```
<AxesSubplot: xlabel='Height', ylabel='Count'>
```



In [13]:
```
sn.histplot(df.Height, kde=False) #kde is for getting the bell curve
```

Out[13]:
```
<AxesSubplot: xlabel='Height', ylabel='Count'>
```



In [13]:
```
mean =df.Height.mean() #for finding the mean
mean
```

Out[13]: 66.367559754866

In [ ]:

In [14]:
```
#range of standard deviation is -3 and +3
std_deviation = df.Height.std()
std_deviation
```

Out[14]: 3.847528120795573

In [15]:
```
mean-3*std_deviation
```

Out[15]: 54.824975392479274

In [16]:
```
#outliers is beyond -3 and +3
mean+3*std_deviation
```

Out[16]: 77.91014411725271

In [14]:
```
df[(df.Height<54.82)|(df.Height>77.91)] #these are outliers
```

Out[14]:

	Gender	Height
994	Male	78.095867
1317	Male	78.462053
2014	Male	78.998742
3285	Male	78.528210
3757	Male	78.621374
6624	Female	54.616858
9285	Female	54.263133

In [15]:
```
#to find no of outliers
df_no_outlier = df[(df.Height>54.82)&(df.Height<77.91)] #for no outliers
df_no_outlier.shape #2d array #9993 are the no of data samples and there are 7 outliers therefore 10000 data samples.
```

Out[15]: (9993, 2)

In [16]:
```
df['zscore']=(df.Height-df.Height.mean())/df.Height.std() #for zscore for standardisation
df.head()#for displaying the table
```

Out[16]:

	Gender	Height	zscore
0	Male	73.847017	1.943964
1	Male	68.781904	0.627505
2	Male	74.110105	2.012343
3	Male	71.730978	1.393991
4	Male	69.881796	0.913375

In [20]:
```
df.Height.mean()
```

Out[20]:
```
66.367559754866
```

In [21]:
```
df.Height.std()
```

Out[21]:
```
3.847528120795573
```

In [22]:
```
(73.847017-66.367559754866)/3.847528120795573
```

Out[22]:
```
1.9439642831219734
```

In [17]:
```
df[(df.zscore<-3)|(df.zscore>3)]
```

Out[17]:

	Gender	Height	zscore
994	Male	78.095867	3.048271
1317	Male	78.462053	3.143445
2014	Male	78.998742	3.282934
3285	Male	78.528210	3.160640
3757	Male	78.621374	3.184854
6624	Female	54.616858	-3.054091
9285	Female	54.263133	-3.146027

In [24]:
```
df_no_outlier=df[(df.zscore>-3)&(df.zscore<3)]
```

```
df_no_outlier.shape
```

Out[24]:
```
(9993, 3)
```

In [18]:
```
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline
matplotlib.rcParams['figure.figsize']=(12,8)
#param is a parameter
#matplotlib inline is for plotting graphs such as scatterplot
```

In [19]:
```
df=pd.read_csv('bhp.csv')
df.head()
```

Out[19]:

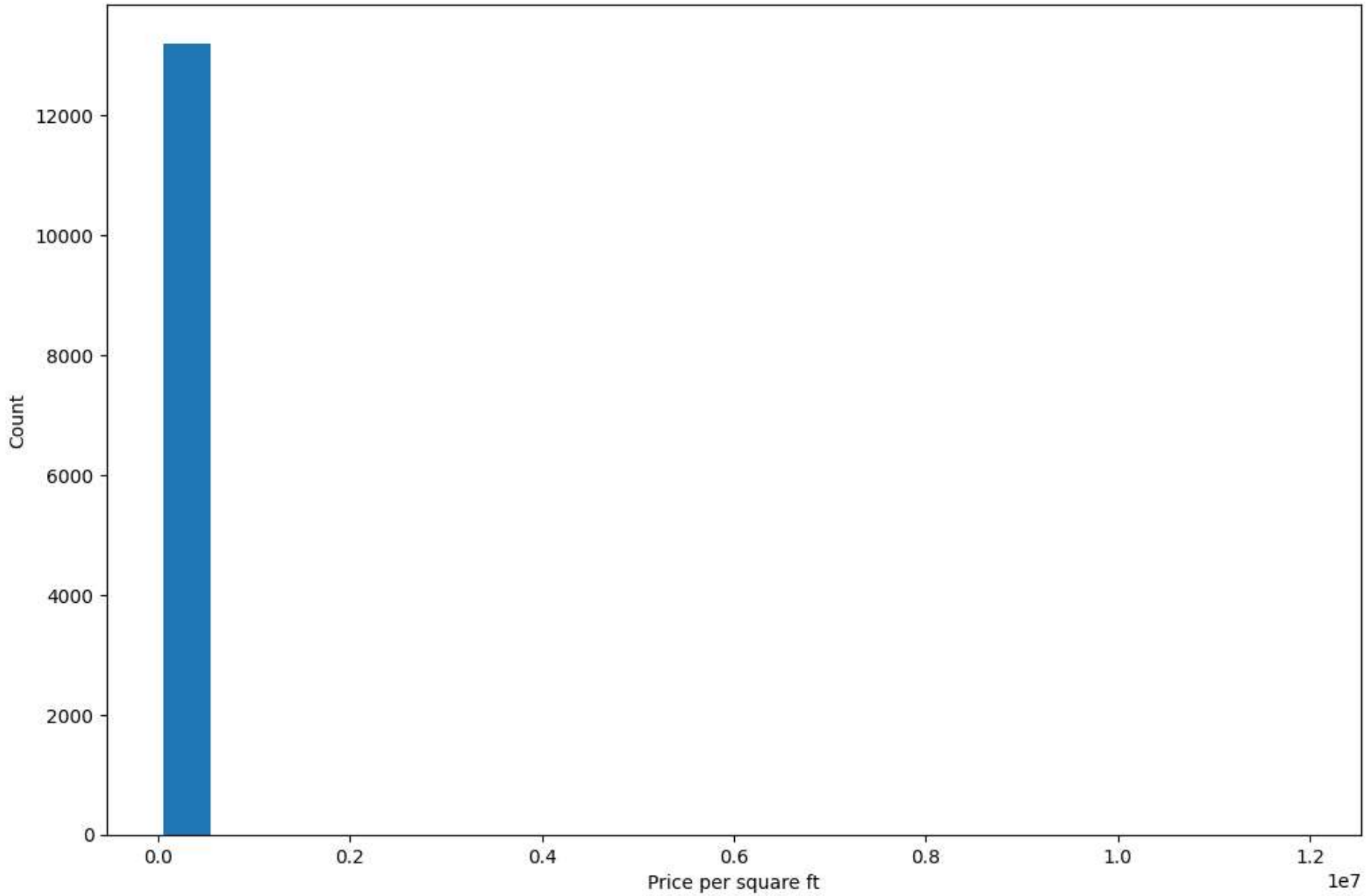
	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250

In [20]:
```
df.price_per_sqft.describe() #for findinf mean min...
```

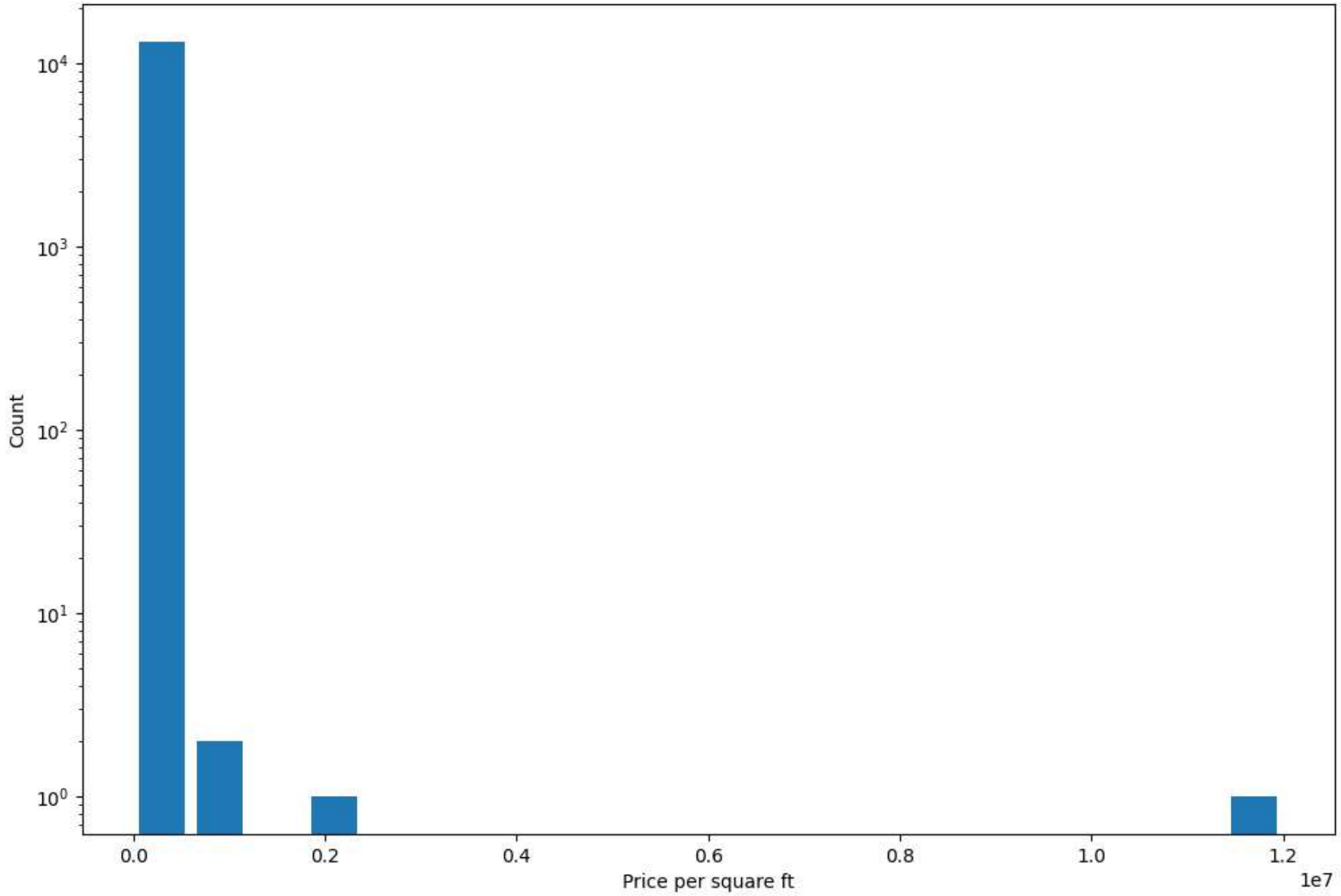
Out[20]:
```
count    1.320000e+04
mean     7.920337e+03
std      1.067272e+05
min      2.670000e+02
25%      4.267000e+03
50%      5.438000e+03
75%      7.317000e+03
max      1.200000e+07
Name: price_per_sqft, dtype: float64
```

In [25]:
```
plt.hist(df.price_per_sqft, bins=20, rwidth=0.8)
#rwidth : This parameter is an optional parameter and it is a relative width of the bars as a fraction of the bin width.
plt.xlabel('Price per square ft')
```

```
plt.ylabel('Count')
plt.show()
#The towers or bars of a histogram are called bins.
#The height of each bin shows how many values from that data fall into that range.
#Width of each bin is = (max value of data - min value of data) / total number of bins
```



In [26]:
```
plt.hist(df.price_per_sqft, bins=20, rwidth=0.8)
plt.xlabel('Price per square ft')
plt.ylabel('Count')
plt.yscale('log')
plt.show()
```



In [31]:
```
#Treat outliers using percentile first
lower_limit,upper_limit =df.price_per_sqft.quantile([0.001,0.999])
lower_limit,upper_limit
```

Out[31]: (1366.184, 50959.36200000098)

In [29]:
```
outliers = df[(df.price_per_sqft>upper_limit)|(df.price_per_sqft<lower_limit)]
outliers.sample(10)
```

Out[29]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
7166	Yelahanka	1 Bedroom	26136.0	1.0	150.0	1	573
5417	Ulsoor	4 BHK	36000.0	4.0	450.0	4	1250
7862	JP Nagar	3 BHK	20000.0	3.0	175.0	3	875
1005	other	1 BHK	15.0	1.0	30.0	1	200000
3934	other	1 BHK	1500.0	1.0	19.5	1	1300
2392	other	4 Bedroom	2000.0	3.0	25.0	4	1250
8300	Kengeri	1 BHK	1200.0	1.0	14.0	1	1166
345	other	3 Bedroom	11.0	3.0	74.0	3	672727
7799	other	4 BHK	2000.0	3.0	1063.0	4	53150
12328	other	4 Bedroom	4350.0	8.0	2600.0	4	59770

In [30]:
```
outliers = df[(df.price_per_sqft>upper_limit)|(df.price_per_sqft<lower_limit)]
outliers.sample(5)
```

Out[30]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
7575	other	1 BHK	425.0	1.0	750.0	1	176470
7862	JP Nagar	3 BHK	20000.0	3.0	175.0	3	875
9144	other	4 Bedroom	10961.0	4.0	80.0	4	729
8307	Bannerghatta Road	5 BHK	2500.0	4.0	1400.0	5	56000
798	other	4 Bedroom	10961.0	4.0	80.0	4	729

In [32]:
```
df2= df[(df.price_per_sqft<upper_limit)&(df.price_per_sqft>lower_limit)]
df2.shape
```

Out[32]: (13172, 7)

In [33]:
```
df.head()
```

Out[33]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250

In [35]:
```
df.shape[0]-df2.shape[0] #we removed 28 outliers #shape 0 and shape 1:
```

Out[35]: 28

In [36]:
```
max_limit = df2.price_per_sqft.mean()+ 4*df2.price_per_sqft.std() #not outliers
min_limit = df2.price_per_sqft.mean()- 4*df2.price_per_sqft.std()
max_limit,min_limit
```

Out[36]: (23227.73653589432, -9900.429065502582)

In [43]:
```
df2[(df2.price_per_sqft>max_limit)|(df2.price_per_sqft<min_limit)].sample(10) #these are the outliers
```

Out[43]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9419	HSR Layout	9 Bedroom	1200.0	9.0	350.0	9	29166
9229	1st Phase JP Nagar	4 Bedroom	1200.0	4.0	300.0	4	25000
6391	other	5 Bedroom	4000.0	4.0	1000.0	5	25000
9711	Rajaji Nagar	2 Bedroom	1056.0	1.0	250.0	2	23674
849	other	4 Bedroom	2400.0	4.0	640.0	4	26666
3665	Koramangala	4 Bedroom	2400.0	6.0	600.0	4	25000
1770	other	10 Bedroom	1660.0	10.0	475.0	10	28614
3144	other	5 BHK	8321.0	5.0	2700.0	5	32448
12352	other	6 Bedroom	2400.0	5.0	750.0	6	31250
3873	other	5 Bedroom	1250.0	5.0	300.0	5	24000

In [44]:
```
df.head(20)
```

Out[44]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828
9	other	6 Bedroom	1020.0	6.0	370.00	6	36274
10	Whitefield	3 BHK	1800.0	2.0	70.00	3	3888
11	Whitefield	4 Bedroom	2785.0	5.0	295.00	4	10592
12	7th Phase JP Nagar	2 BHK	1000.0	2.0	38.00	2	3800
13	Gottigere	2 BHK	1100.0	2.0	40.00	2	3636
14	Sarjapur	3 Bedroom	2250.0	3.0	148.00	3	6577
15	Mysore Road	2 BHK	1175.0	2.0	73.50	2	6255
16	Bisuvanahalli	3 BHK	1180.0	3.0	48.00	3	4067
17	Raja Rajeshwari Nagar	3 BHK	1540.0	3.0	60.00	3	3896
18	other	3 BHK	2770.0	4.0	290.00	3	10469
19	other	2 BHK	1100.0	2.0	48.00	2	4363

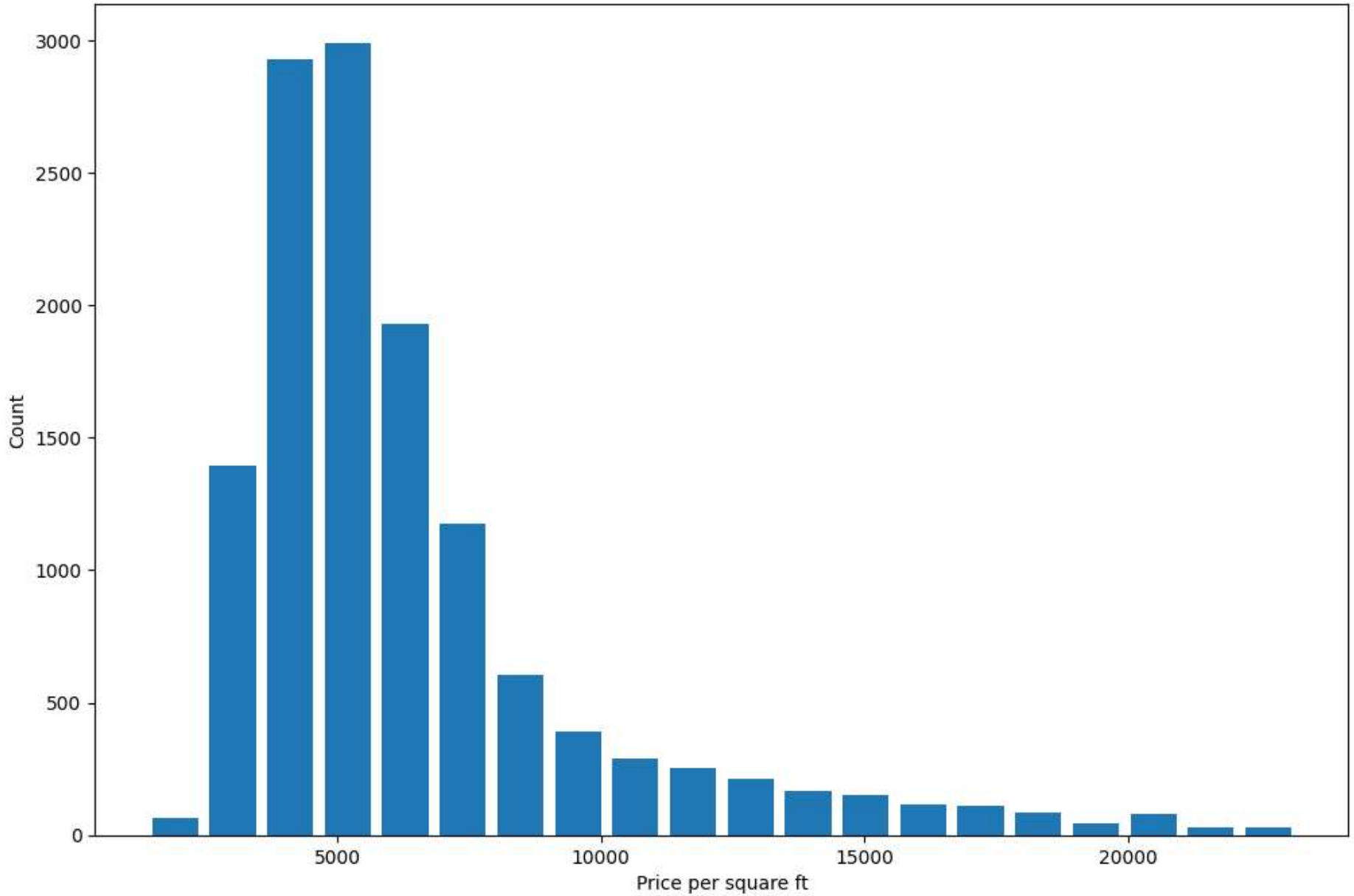
In [45]:
```
df3= df2[(df2.price_per_sqft>min_limit)&(df2.price_per_sqft<max_limit)]
df3.shape
```

Out[45]:
```
(13047, 7)
```

In [46]:
```
df2.shape[0]-df3.shape[0]#in this we removed total 125 outliers
```

Out[46]:
```
125
```

In [47]:
```
plt.hist(df3.price_per_sqft, bins=20, rwidth=0.8)
plt.xlabel('Price per square ft')
plt.ylabel('Count')
plt.show()
```



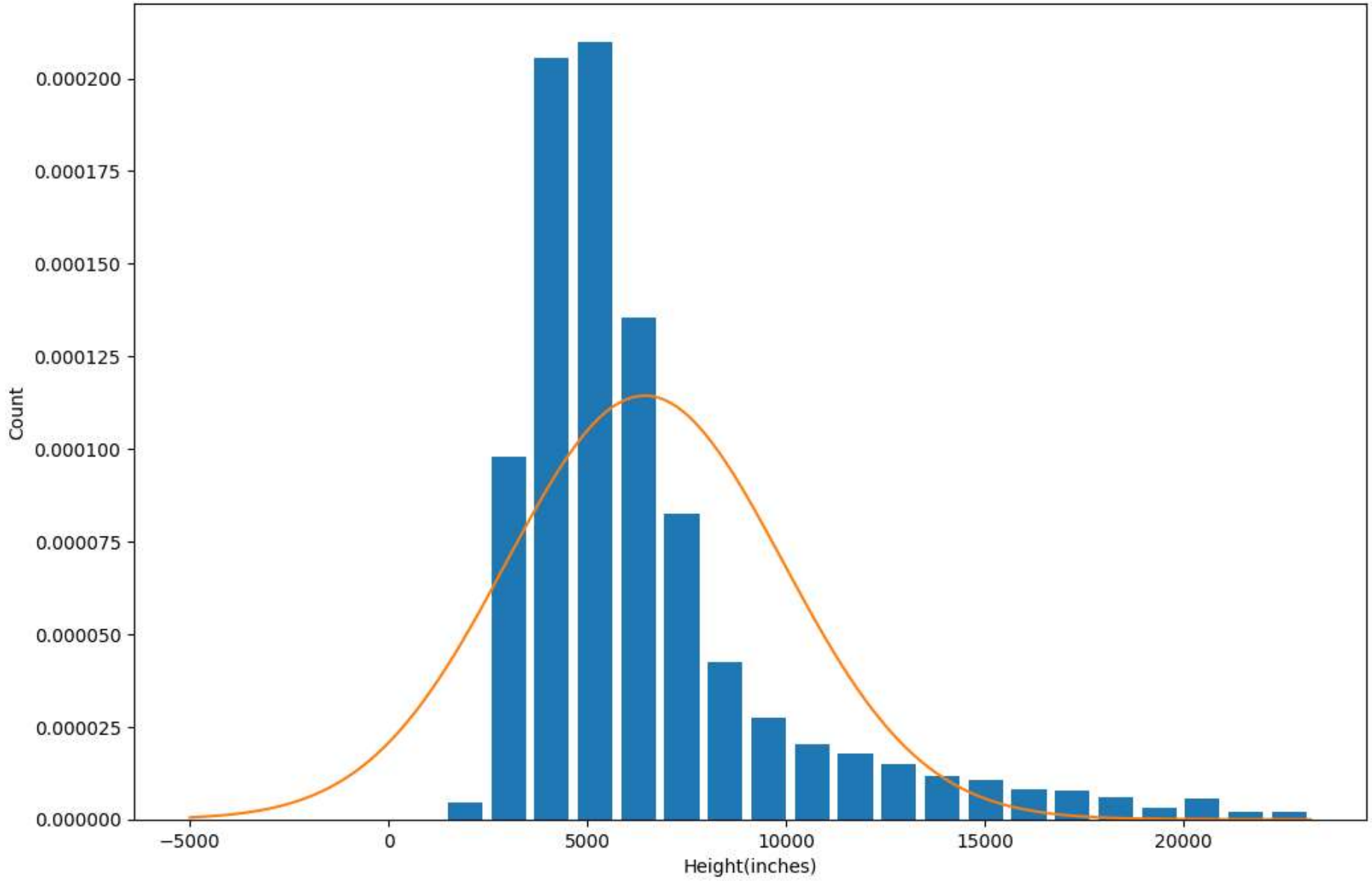
In [ ]:

In [52]:
```
from scipy.stats import norm
import numpy as np
```

```
plt.hist(df3.price_per_sqft, bins=20, rwidth=0.8, density=True)
plt.xlabel('Height(inches)')
plt.ylabel('Count')

rng = np.arange(-5000,df3.price_per_sqft.max(), 100) #range
plt.plot(rng, norm.pdf(rng,df3.price_per_sqft.mean(),df3.price_per_sqft.std())) #arange (start, stop, step)
#Values are generated within the half-open interval [start, stop), with spacing between values given by step.
```

Out[52]: [<matplotlib.lines.Line2D at 0x14f2de5c1c0>]



In [54]:
```
#Now remove outliers using zscore. using zscore of 4 as your threshold
df2['zscore'] = (df2.price_per_sqft-df2.price_per_sqft.mean())/df2.price_per_sqft.std()
df2.sample(10)
```

```
C:\Users\Anusha\AppData\Local\Temp\ipykernel_6040\2930407720.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2['zscore'] = (df2.price_per_sqft-df2.price_per_sqft.mean())/df2.price_per_sqft.std()
```

Out[54]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft	zscore
7721	other	3 BHK	1310.0	2.0	55.0	3	4198	-0.595422
8208	Hennur Road	2 BHK	1232.0	2.0	87.0	2	7061	0.095954
2735	5th Phase JP Nagar	2 BHK	1450.0	2.0	58.0	2	4000	-0.643236
11962	other	2 BHK	600.0	4.0	70.0	2	11666	1.207998
6232	Abbigere	4 Bedroom	1200.0	3.0	120.0	4	10000	0.805682
3185	Begur	3 BHK	1559.0	3.0	63.0	3	4041	-0.633335
9482	Old Airport Road	3 BHK	2392.0	3.0	249.0	3	10409	0.904450
8424	Electronic City	3 BHK	1050.0	2.0	39.0	3	3714	-0.712301
875	Hosur Road	5 Bedroom	3300.0	5.0	240.0	5	7272	0.146907
2858	Kanakpura Road	3 BHK	1450.0	3.0	56.0	3	3862	-0.676561

In [55]:
```
df3['zscore'] = (df3.price_per_sqft-df3.price_per_sqft.mean())/df3.price_per_sqft.std()
df3.sample(10)
```

```
C:\Users\Anusha\AppData\Local\Temp\ipykernel_6040\2992638218.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df3['zscore'] = (df3.price_per_sqft-df3.price_per_sqft.mean())/df3.price_per_sqft.std()
```

Out[55]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft	zscore
4750	Chandapura	3 BHK	1190.0	2.0	30.35	3	2550	-1.118032
2120	Kammanahalli	2 BHK	1160.0	2.0	52.00	2	4482	-0.564081
3822	other	3 BHK	1310.0	2.0	37.83	3	2887	-1.021406
2189	other	2 BHK	1140.0	2.0	76.00	2	6666	0.062125
7567	Kundalahalli	3 BHK	1496.0	2.0	78.00	3	5213	-0.354485
3671	Somasundara Palya	2 BHK	1033.0	2.0	48.00	2	4646	-0.517058
7334	other	2 BHK	1475.0	2.0	70.00	2	4745	-0.488672
7661	Thanisandra	3 BHK	1795.0	3.0	150.00	3	8356	0.546689
5432	Kanakpura Road	2 BHK	700.0	2.0	34.99	2	4998	-0.416131
11945	Hoodi	5 Bedroom	3250.0	5.0	395.00	5	12153	1.635382

In [56]:
```
outliers_z = df2[(df2.zscore<-4)|(df2.zscore>4)]
outliers_z.shape
```

Out[56]:
```
(125, 8)
```

In [57]:
```
outliers_z.sample(5)
```

Out[57]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft	zscore
8082	Nagarbhavi	4 Bedroom	1200.00	3.0	340.0	4	28333	5.232851
10000	other	6 Bedroom	1200.00	5.0	280.0	6	23333	4.025420
4357	other	4 Bedroom	3250.00	5.0	850.0	4	26153	4.706411
12640	other	3 BHK	2777.29	5.0	649.0	3	23368	4.033872
3675	Kasturi Nagar	5 Bedroom	1650.00	5.0	450.0	5	27272	4.976634

In [58]:
```
df4= df2[(df2.zscore>-4)&(df2.zscore<4)]
df4.shape
```

Out[58]: (13047, 8)

In [59]:
```
df2.shape[0]-df4.shape[0]
```

Out[59]: 125

.shape:

example:

outlier:

example:

In [ ]: