In [1]:

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

In [4]:

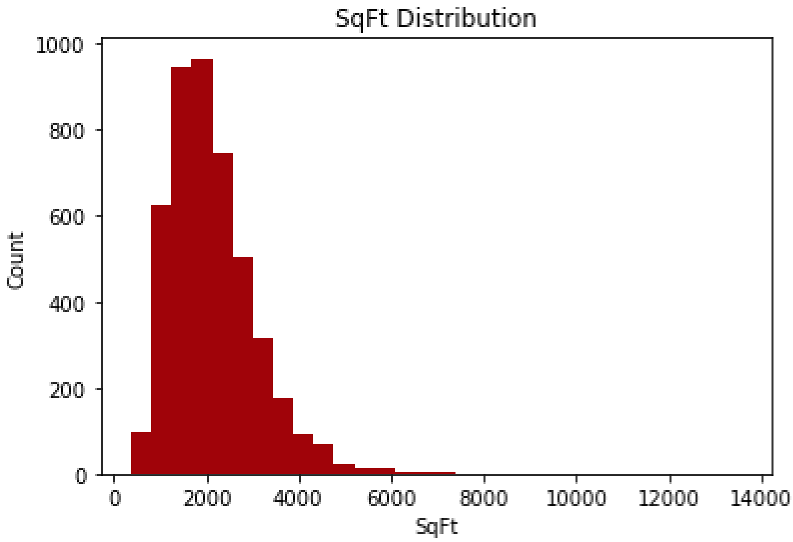
```
df = pd.read_csv('../data.csv')
```

In [18]:

```
sq=df['sqft_living']
plt.hist(sq,bins=30, color = '#a00309')
plt.title('SqFt Distribution')
plt.xlabel('SqFt')
plt.ylabel('Count')
```

Out[18]:

```
Text(0, 0.5, 'Count')
```

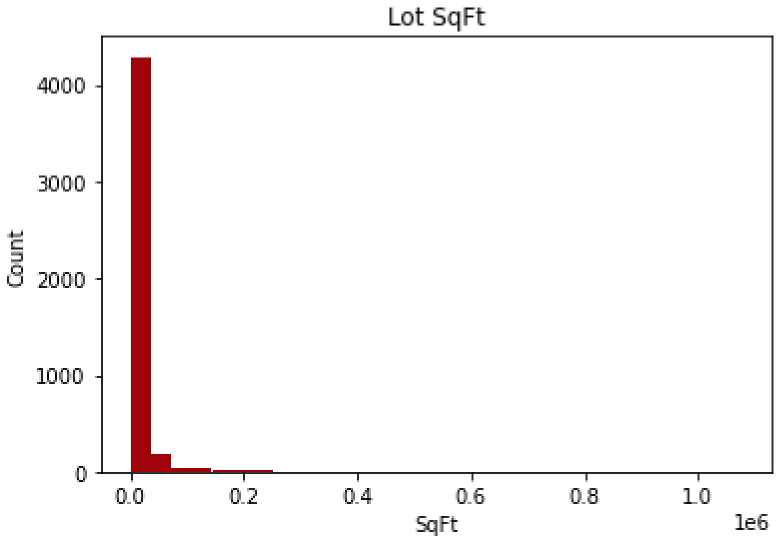


In [25]:

```python
sq=df['sqft_lot']
plt.hist(sq,bins=30, color = '#a00309')
plt.title('Lot SqFt')
plt.xlabel('SqFt')
plt.ylabel('Count')
```

Out[25]:

```
Text(0, 0.5, 'Count')
```



In [26]:

```
df
```

Out[26]:

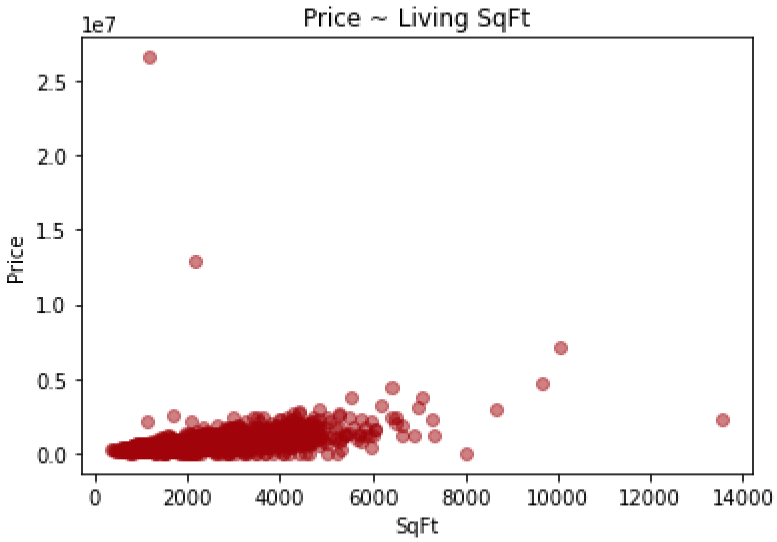
athrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	y
1.50	1340	7912	1.5	0	0	3	1340	0	
2.50	3650	9050	2.0	0	4	5	3370	280	
2.00	1930	11947	1.0	0	0	4	1930	0	
2.25	2000	8030	1.0	0	0	4	1000	1000	
2.50	1940	10500	1.0	0	0	4	1140	800	
...	...	...	...	...	...	...	...	...	
1.75	1510	6360	1.0	0	0	4	1510	0	
2.50	1460	7573	2.0	0	0	3	1460	0	
2.50	3010	7014	2.0	0	0	3	3010	0	
2.00	2090	6630	1.0	0	0	3	1070	1020	
2.50	1490	8102	2.0	0	0	4	1490	0	

In [31]:

```
price=df['price']
sq=df['sqft_living']
plt.scatter(x=sq,y=price, color = '#a00309', alpha=.5)
plt.title('Price ~ Living SqFt')
plt.xlabel('SqFt')
plt.ylabel('Price')
```

Out[31]:

```
Text(0, 0.5, 'Price')
```



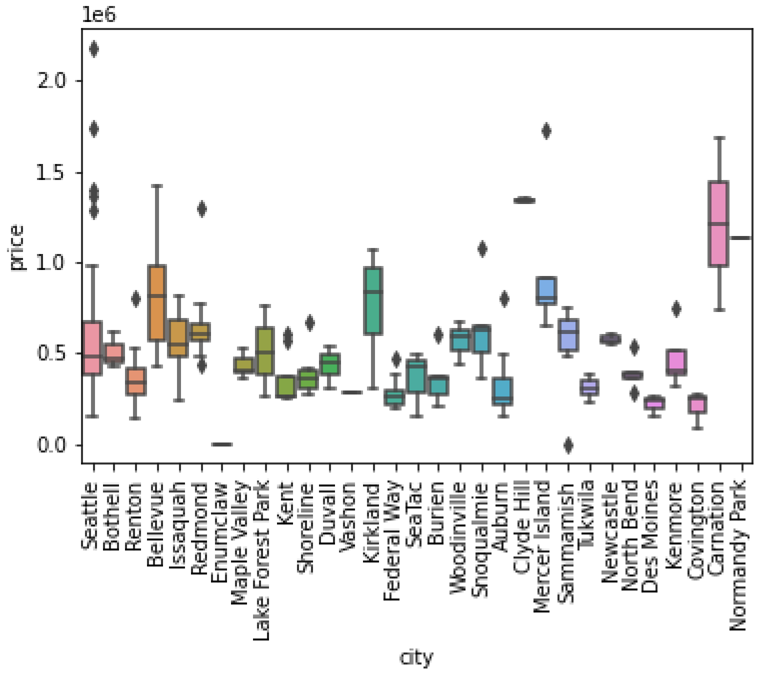
In [55]:

```
box=df[['price','city']]
import seaborn as sns
box=box.sample(n=300)
ax = sns.boxplot(x="city", y="price", data=box)
plt.xticks(rotation=90)

```

Out[55]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]),
 [Text(0, 0, 'Seattle'),
  Text(1, 0, 'Bothell'),
  Text(2, 0, 'Renton'),
  Text(3, 0, 'Bellevue'),
  Text(4, 0, 'Issaquah'),
  Text(5, 0, 'Redmond'),
  Text(6, 0, 'Enumclaw'),
  Text(7, 0, 'Maple Valley'),
  Text(8, 0, 'Lake Forest Park'),
  Text(9, 0, 'Kent'),
  Text(10, 0, 'Shoreline'),
  Text(11, 0, 'Duvall'),
  Text(12, 0, 'Vashon'),
  Text(13, 0, 'Kirkland'),
  Text(14, 0, 'Federal Way'),
  Text(15, 0, 'SeaTac'),
  Text(16, 0, 'Burien'),
  Text(17, 0, 'Woodinville'),
  Text(18, 0, 'Snoqualmie'),
  Text(19, 0, 'Auburn'),
  Text(20, 0, 'Clyde Hill'),
  Text(21, 0, 'Mercer Island'),
  Text(22, 0, 'Sammamish'),
  Text(23, 0, 'Tukwila'),
  Text(24, 0, 'Newcastle'),
  Text(25, 0, 'North Bend'),
  Text(26, 0, 'Des Moines'),
  Text(27, 0, 'Kenmore'),
  Text(28, 0, 'Covington'),
  Text(29, 0, 'Carnation'),
  Text(30, 0, 'Normandy Park')])
```



In [119]:

```
df=pd.read_excel('../ZestAccuracy.xlsx')
```

In [120]:

```
df
```

21	Houston, TX	Market	--	24.0K	--	--	--
22	Kansas City, MO	Off-Market	--	688.6K	--	--	--
23	Kansas City, MO	On-Market	--	11.7K	--	--	--
24	Los Angeles-Long Beach-Anaheim, CA	Off-Market	6.1%	2.9M	43.2%	67.7%	86.8%
25	Los Angeles-Long Beach-Anaheim, CA	On-Market	1.7%	32.1K	85.5%	96.3%	99.3%
26	Miami-Fort Lauderdale, FL	Off-Market	7.3%	2.2M	37.4%	61.2%	82.6%
27	Miami-Fort Lauderdale, FL	On-Market	2.3%	59.0K	82.4%	95.1%	98.9%
28	Minneapolis-St Paul, MN	Off-Market	6.4%	1.2M	41.2%	66.1%	86.0%
29	Minneapolis-St Paul, MN	On-Market	1.8%	17.3K	86.3%	97.0%	99.4%

In [121]:

```
df=df[['Msa', 'Type','Median Error', 'Within 5% of Sale Price']]
```

In [122]:

```
df['Median Error']=df['Median Error'].str[:-1]
df['Within 5% of Sale Price']=df['Within 5% of Sale Price'].str[:-1]

df['Median Error']=pd.to_numeric(df['Median Error'], errors='coerce')
df['Within 5% of Sale Price']=pd.to_numeric(df['Within 5% of Sale Price'
```

```
C:\Users\CONNOR~1\AppData\Local\Temp/ipykernel_43656/1064329460.py:1: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df['Median Error']=df['Median Error'].str[:-1]
C:\Users\CONNOR~1\AppData\Local\Temp/ipykernel_43656/1064329460.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df['Within 5% of Sale Price']=df['Within 5% of Sale Price'].str[:-1]
C:\Users\CONNOR~1\AppData\Local\Temp/ipykernel_43656/1064329460.py:4: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df['Median Error']=pd.to_numeric(df['Median Error'], errors='coerce')
C:\Users\CONNOR~1\AppData\Local\Temp/ipykernel_43656/1064329460.py:5: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df['Within 5% of Sale Price']=pd.to_numeric(df['Within 5% of Sale Pric
e'], errors='coerce')
```

In [123]:

```
df1=df.loc[df.Type=='On-Market']
```

In [ ]:

```

```

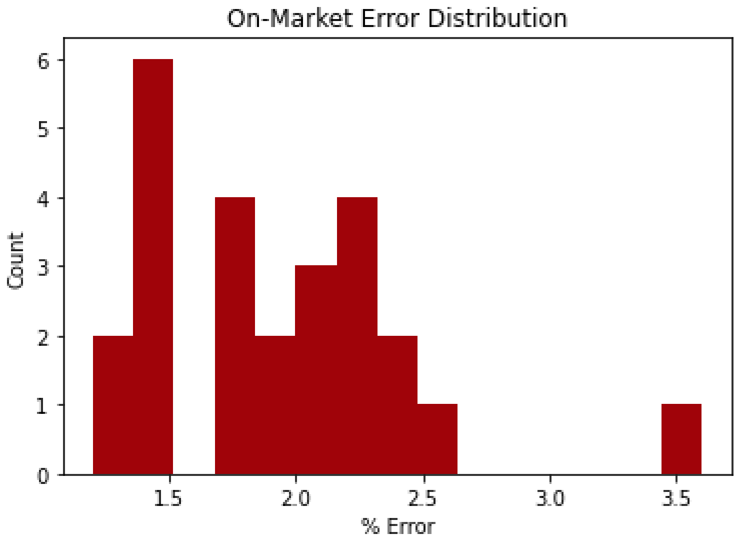
In [124]:

```

me=df1['Median Error']
plt.hist(me,bins=15, color = '#a00309')
plt.title('On-Market Error Distribution')
plt.xlabel('% Error')
plt.ylabel('Count')
```

Out[124]:

```
Text(0, 0.5, 'Count')
```



In [125]:

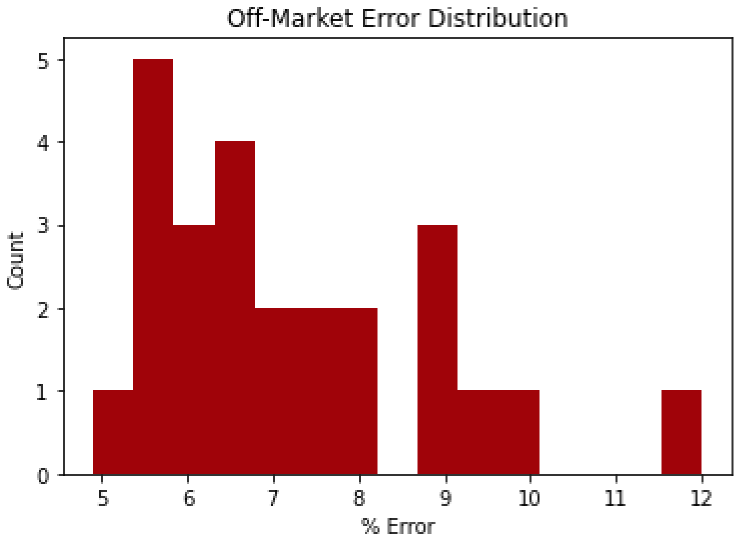
```
df2=df.loc[df.Type=='Off-Market']
```

In [126]:

```
me=df2['Median Error']
plt.hist(me,bins=15, color = '#a00309')
plt.title('Off-Market Error Distribution')
plt.xlabel('% Error')
plt.ylabel('Count')
```

Out[126]:

```
Text(0, 0.5, 'Count')
```

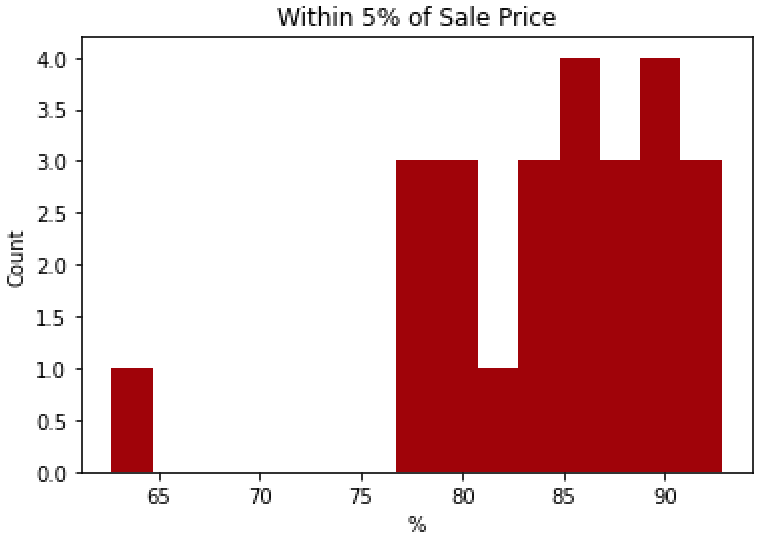


In [127]:

```
me=df1['Within 5% of Sale Price']
plt.hist(me,bins=15, color = '#a00309')
plt.title('Within 5% of Sale Price')
plt.xlabel('%')
plt.ylabel('Count')
```

Out[127]:

```
Text(0, 0.5, 'Count')
```



In [ ]:

```

```