Importing Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
sns.set_style('whitegrid')
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('data_preprocessed.csv')
```

Overview

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Unnamed: 0', 'Price', 'Area', 'Location', 'No. of Bedrooms', 'Resale',
       'ShoppingMall', 'ATM', 'School', 'Hospital', 'VaastuCompliant', 'City',
       'Amenities', 'Furnitioned', 'Capacity(no. of people)'],
      dtype='object')
```

In [5]:

```
df.head()
```

Out[5]:

	Unnamed: 0	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amen
0	0	105.0	1200	Sector 10 Dwarka	2.0	1	0.0	0.0	0.0	0.0	1.0	Delhi	
1	1	60.0	1000	Uttam Nagar	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	
2	2	150.0	1350	Sarita Vihar	2.0	1	0.0	0.0	0.0	0.0	0.0	Delhi	
3	3	25.0	435	Uttam Nagar	2.0	0	0.0	0.0	0.0	0.0	1.0	Delhi	
4	4	58.0	900	Dwarka Mor	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	

In [7]:

```
df['City'].value_counts()
```

Out[7]:

```
City
Hyderabad    2434
Chennai      2233
Delhi        2001
Bangalore    1951
```

```
Mumbai        1398
Kolkata         75
Name: count, dtype: int64
```

feature generation

The orignal dataset have too many unnecessay feature, so we need to generate new features using this old features

In [8]:

```
tmpdf = df.copy()
```

In [18]:

```
# facilities
# psycometric segment
Amenities = ['MaintenanceStaff', 'Gymnasium', 'SwimmingPool', 'LandscapedGardens',
             'JoggingTrack', 'RainWaterHarvesting', 'IndoorGames',
             'Intercom', 'SportsFacility', 'ClubHouse','24X7Security', 'PowerBackup',
              'CarParking', 'StaffQuarter','Cafeteria', 'MultipurposeRoom', 'Children\'s
playarea', 'LiftAvailable']

furnitioned = ['BED', 'Microwave', 'TV', 'DiningTable', 'Sofa', 'Refrigerator', 'AC', 'W
ashingMachine', 'Gasconnection']
```

In [20]:

```
df['Amenities'] = 0;
for i in Amenities:
  df[i] = 0
  df['Amenities'] += df[i]
  df['Amenities'] =  np.where(df['Amenities'] >= 1, 1, 0)
```

In [22]:

```
df.drop(columns=Amenities, axis=1, inplace=True)
```

In [25]:

```
df['Furnitioned'] = 0
for i in furnitioned:
  df[i] = 0
  df['Furnitioned'] += df[i]
  df['Furnitioned'] = np.where(df['Furnitioned']>=1, 1, 0)
df.drop(columns=furnitioned, axis=1, inplace=True)
```

In [26]:

```
df['Capacity(no. of people)'] = df['No. of Bedrooms'] * 2
```

In [28]:

```
df.to_csv('data_preprocessed.csv')
```

EDA

City

In [29]:

```
df.head()
```

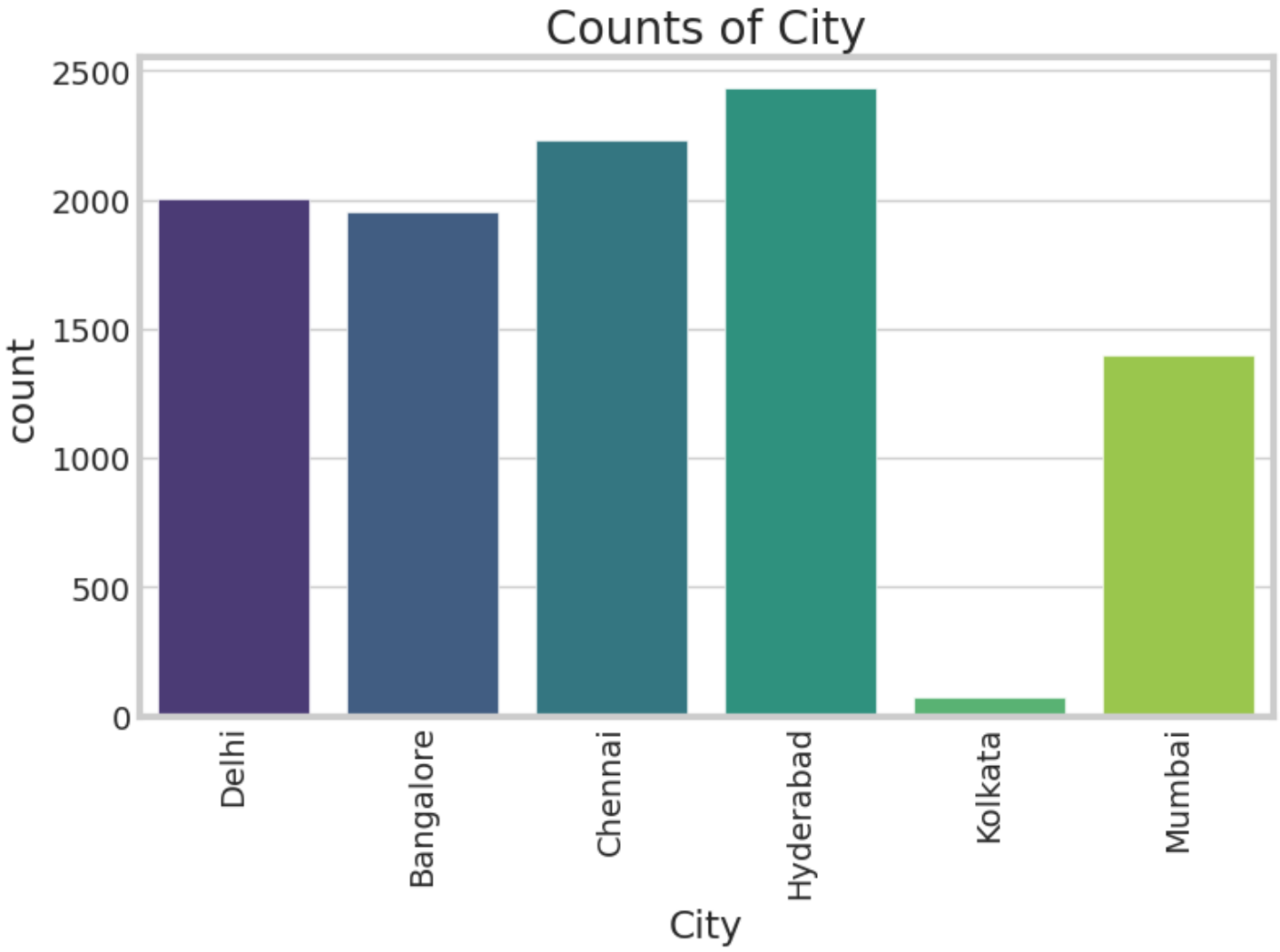
Out[29]:

Unnamed: 0	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amen

	Unnamed: 0	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amen
0	0	105.0	1200	Sector 10 Dwarka	2.0	1	0.0	0.0	0.0	0.0	1.0	Delhi	
1	1	60.0	1000	Uttam Nagar	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	
2	2	150.0	1350	Sarita Vihar	2.0	1	0.0	0.0	0.0	0.0	0.0	Delhi	
3	3	25.0	435	Uttam Nagar	2.0	0	0.0	0.0	0.0	0.0	1.0	Delhi	
4	4	58.0	900	Dwarka Mor	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	

In [39]:

```
plt.figure(figsize=(8, 5))
sns.countplot(x='City', data=df, palette='viridis')
plt.title('Counts of City')
plt.xticks(rotation=90)
plt.show()
```



In [40]:

```
df['City'].value_counts().sort_values(ascending=False)
```

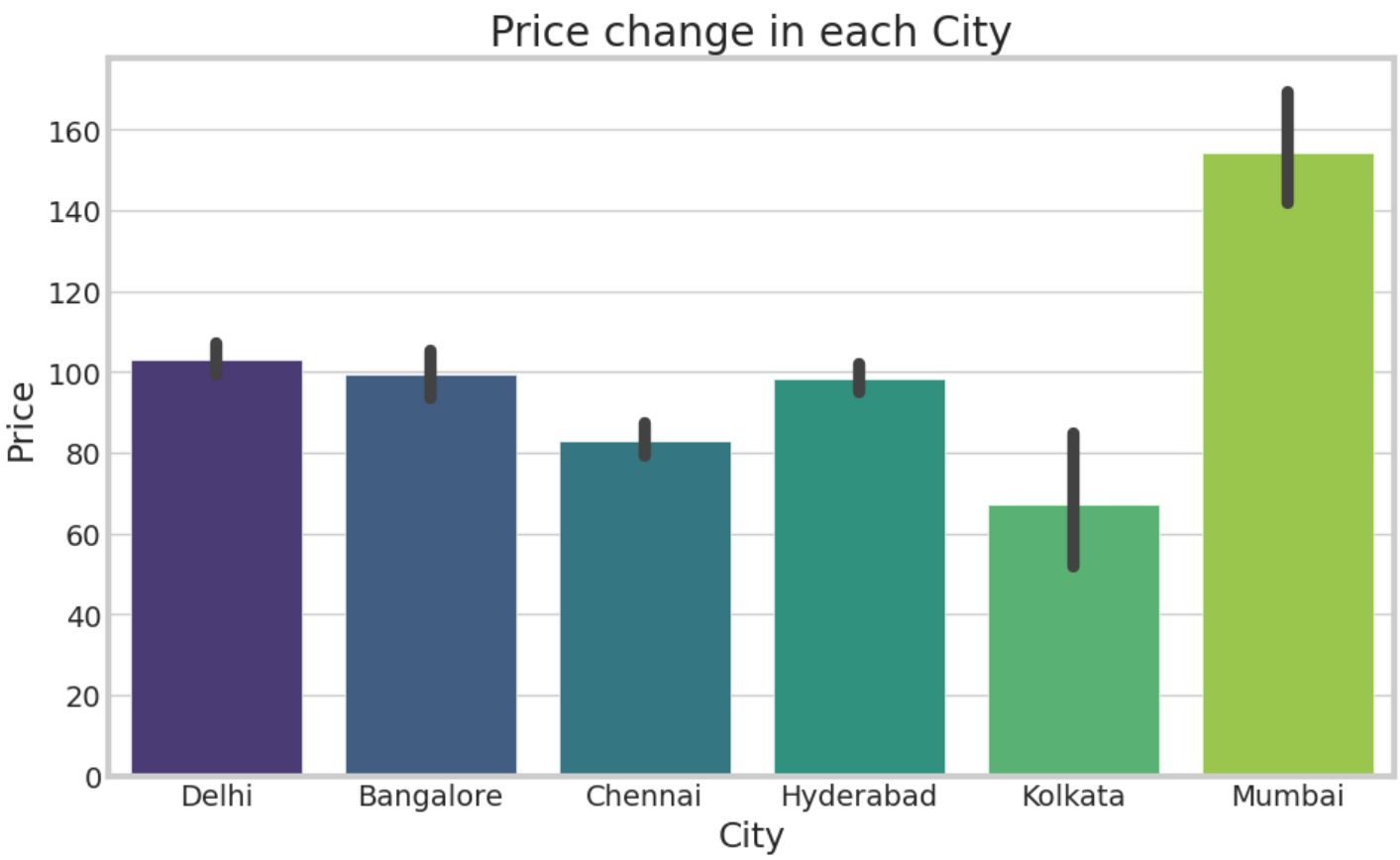
Out[40]:

```
City
Hyderabad    2434
Chennai      2233
Delhi        2001
Bangalore    1951
Mumbai       1398
```

```
Kolkata         75
Name: count, dtype: int64
```

In [44]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x='City', y='Price', data=df, palette='viridis')
plt.title('Price change in each City')
plt.show()
```



In [45]:

```
len(df['Location'].unique())
```

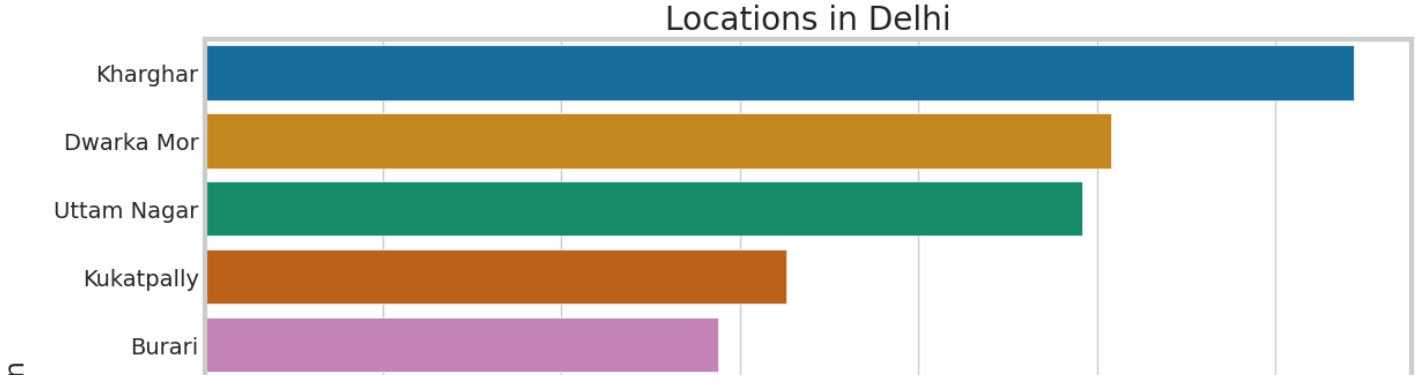
Out[45]:

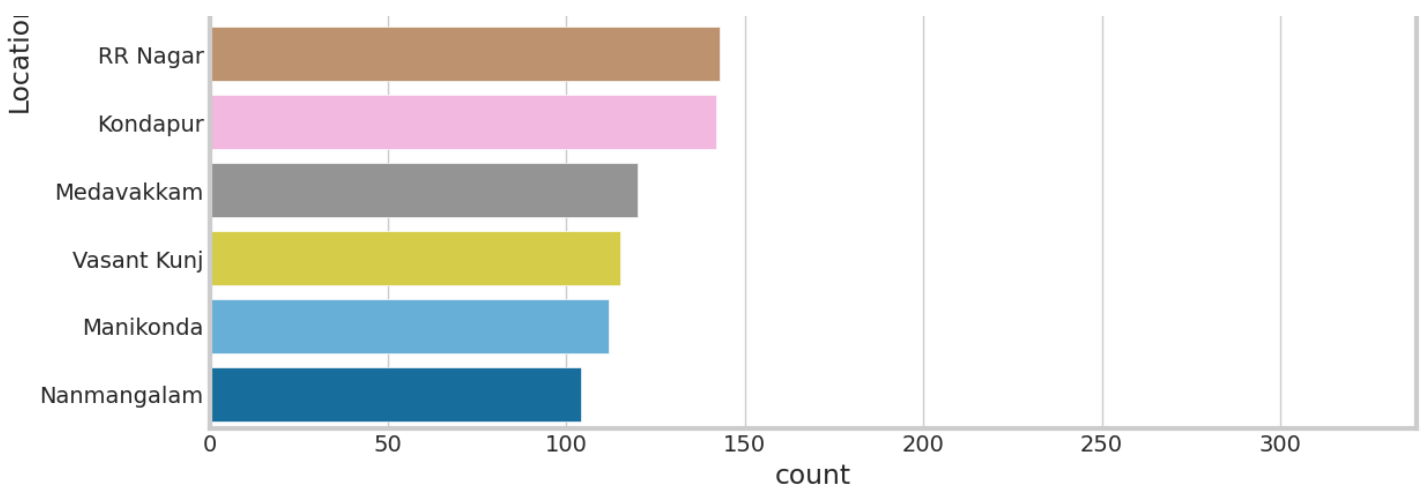
```
849
```

Frequncy of each location in city [top 10]

In [48]:

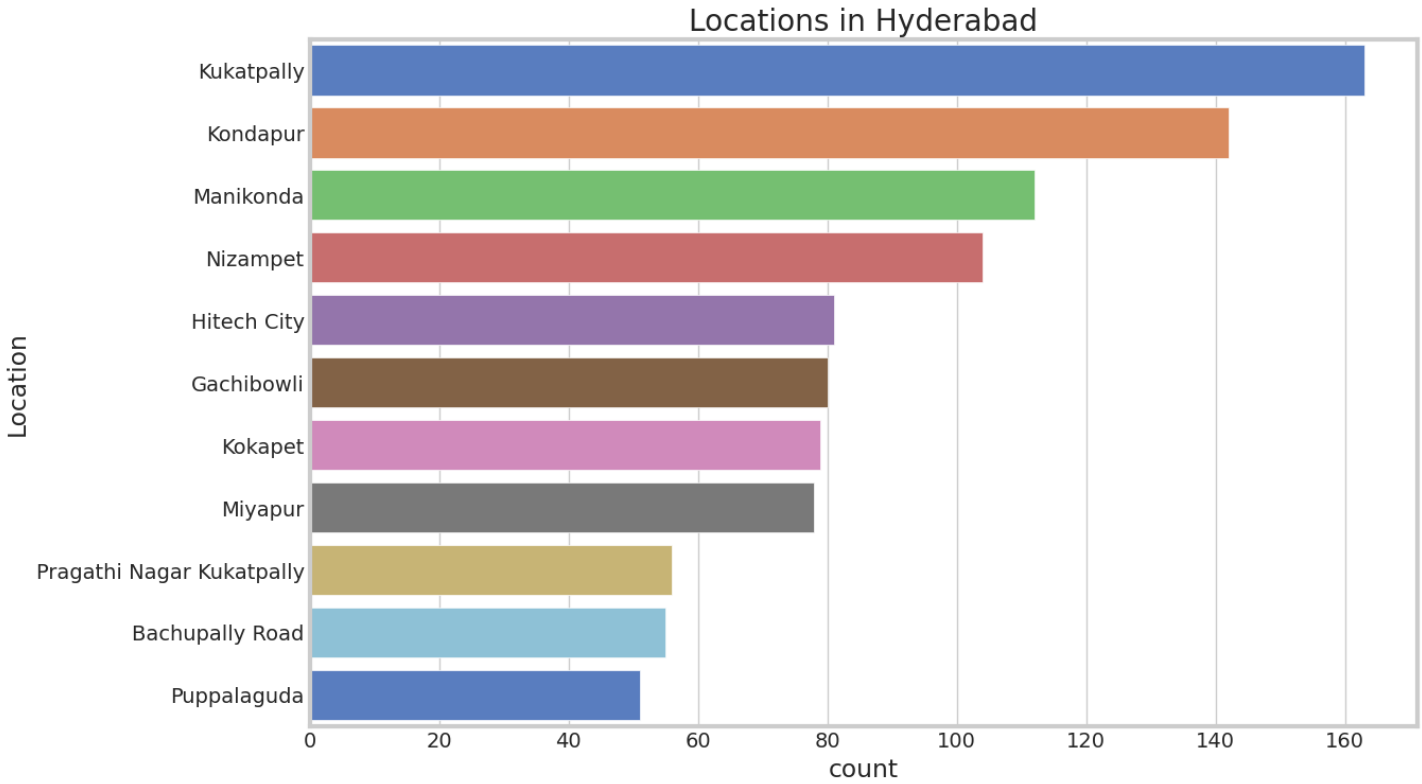
```
mpdf = df[df['City'] == 'Delhi']
plt.figure(figsize=(12, 8))
sns.countplot(y='Location', data=tmpdf, palette='colorblind', order=tmpdf['Location'].va
lue_counts().index[:11])
plt.title('Locations in Delhi')
plt.show()
```





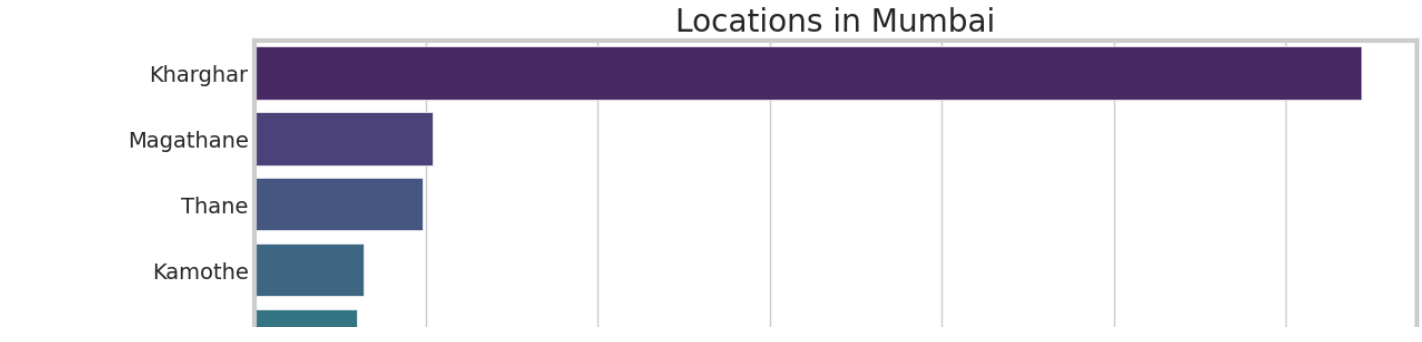
In [50]:

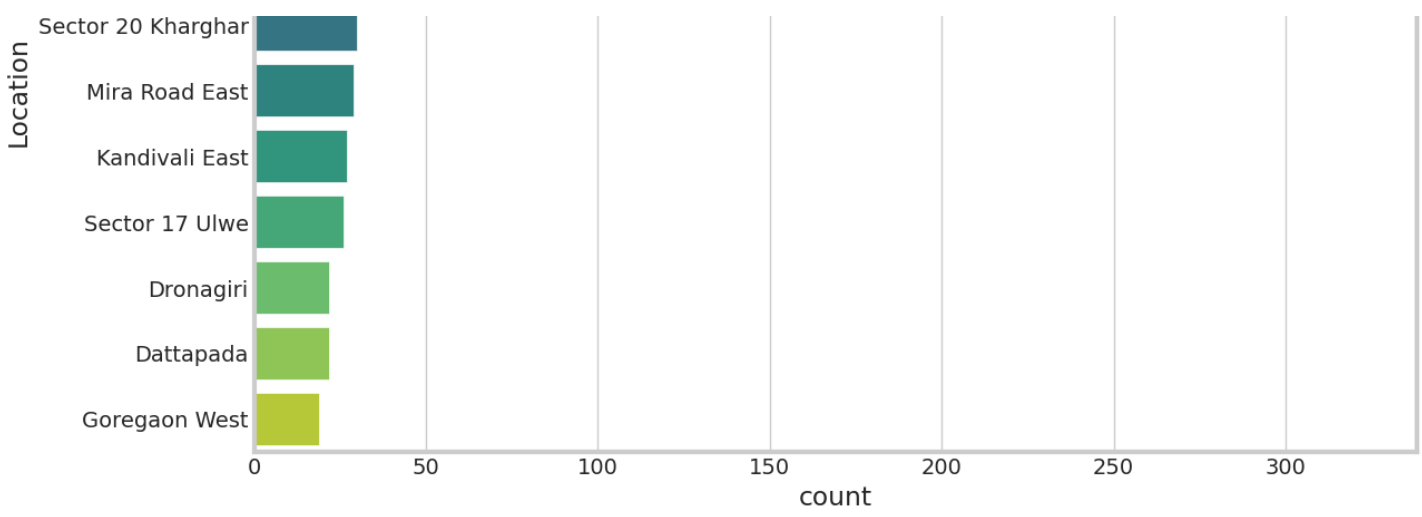
```
tmpdf = df[df['City'] == 'Hyderabad']
plt.figure(figsize=(12, 8))
sns.countplot(y='Location', data=tmpdf, palette='muted', order=tmpdf['Location'].value_c
ounts().index[:11])
plt.title('Locations in Hyderabad')
plt.show()
```



In [52]:

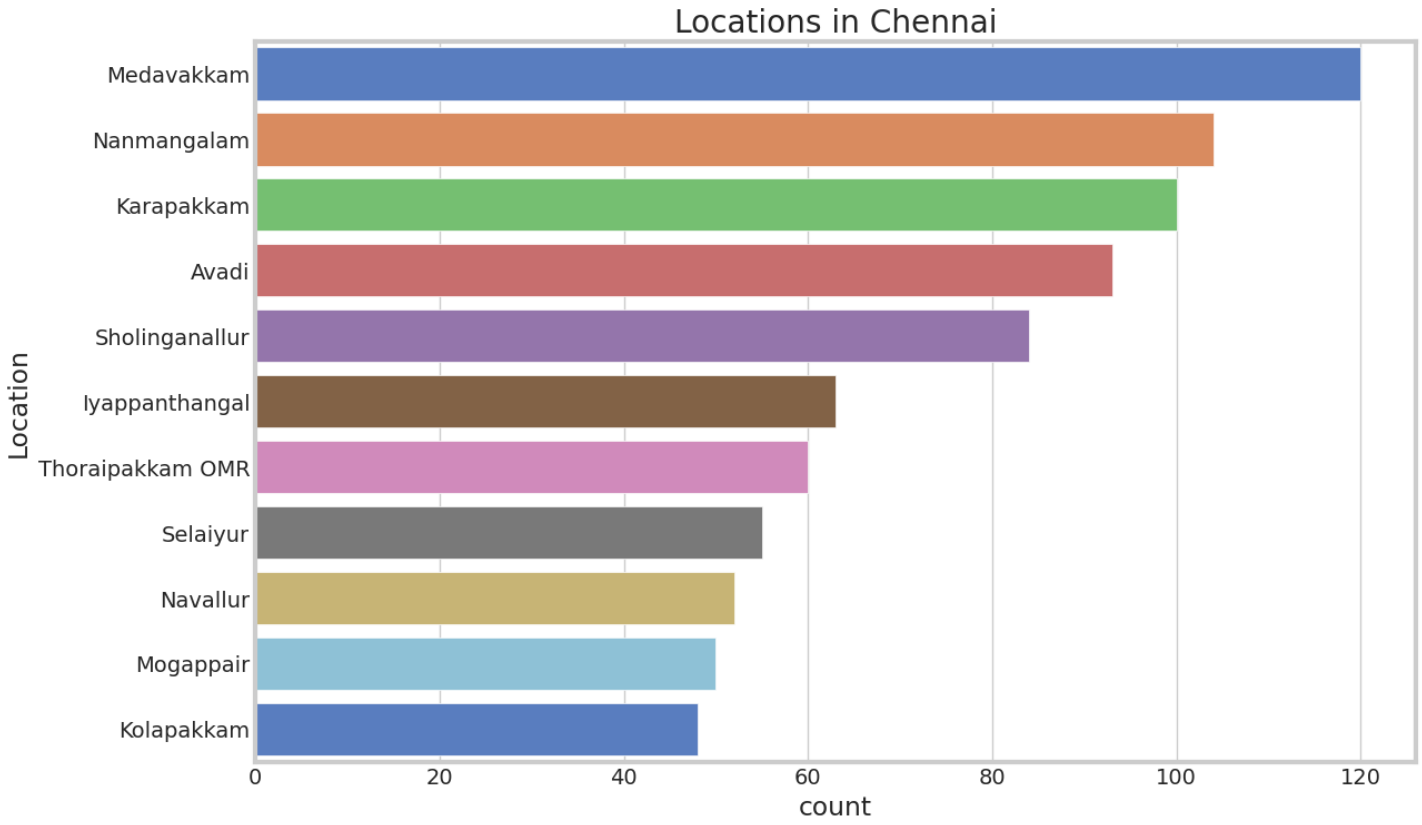
```
tmpdf = df[df['City'] == 'Mumbai']
plt.figure(figsize=(12, 8))
sns.countplot(y='Location', data=tmpdf,palette='viridis', order=tmpdf['Location'].value_
counts().index[:11])
plt.title('Locations in Mumbai')
plt.show()
```





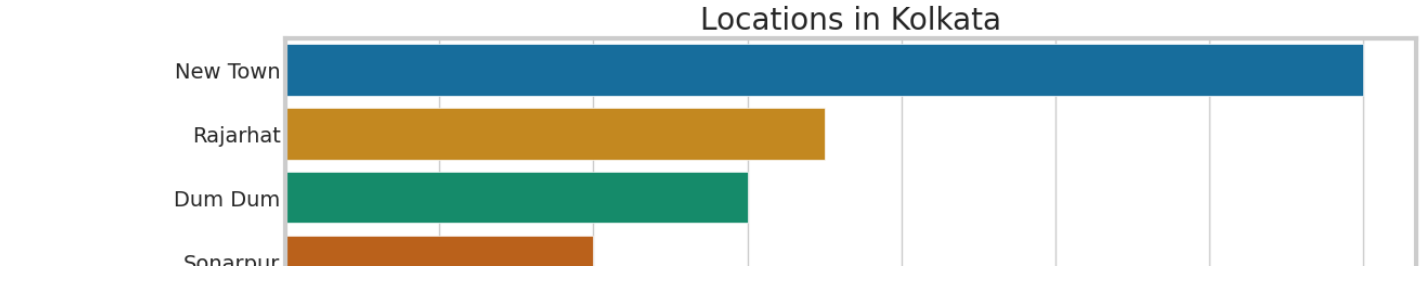
In [59]:

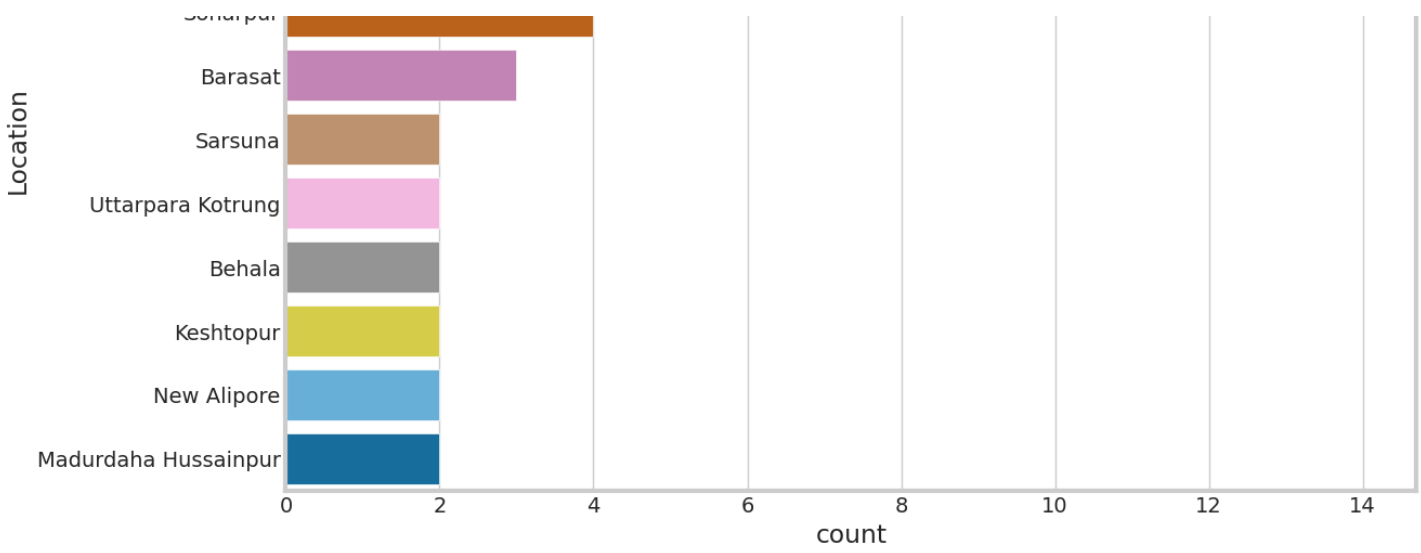
```
tmpdf = df[df['City'] == 'Chennai']
plt.figure(figsize=(12, 8))
sns.countplot(y='Location', data=tmpdf,palette='muted',order=tmpdf['Location'].value_cou
nts().index[:11])
plt.title('Locations in Chennai')
plt.show()
```



In [60]:

```
tmpdf = df[df['City'] == 'Kolkata']
plt.figure(figsize=(12, 8))
sns.countplot(y='Location', data=tmpdf,palette='colorblind',order=tmpdf['Location'].valu
e_counts().index[:11])
plt.title('Locations in Kolkata')
plt.show()
```

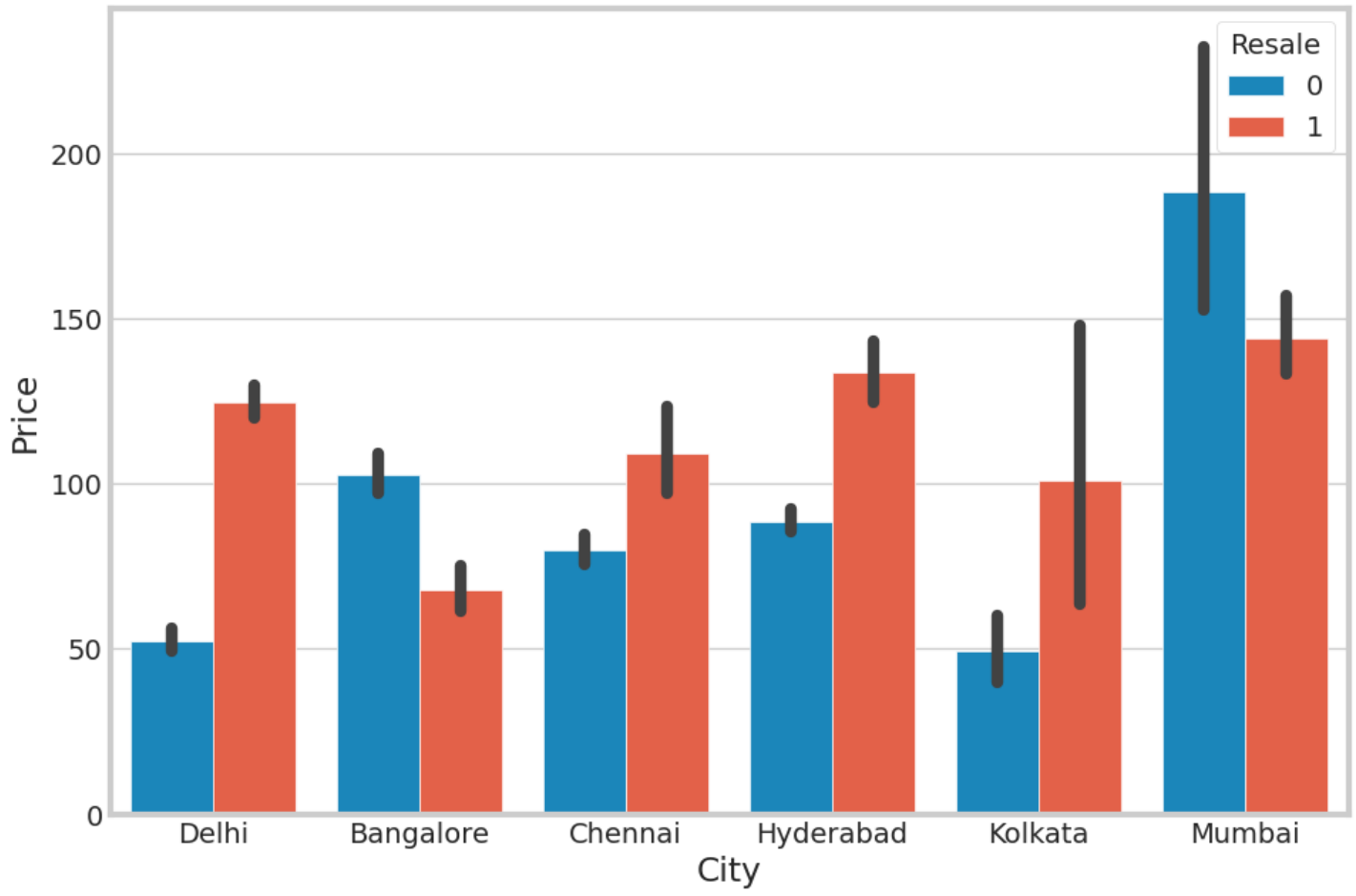




Resale

In [62]:

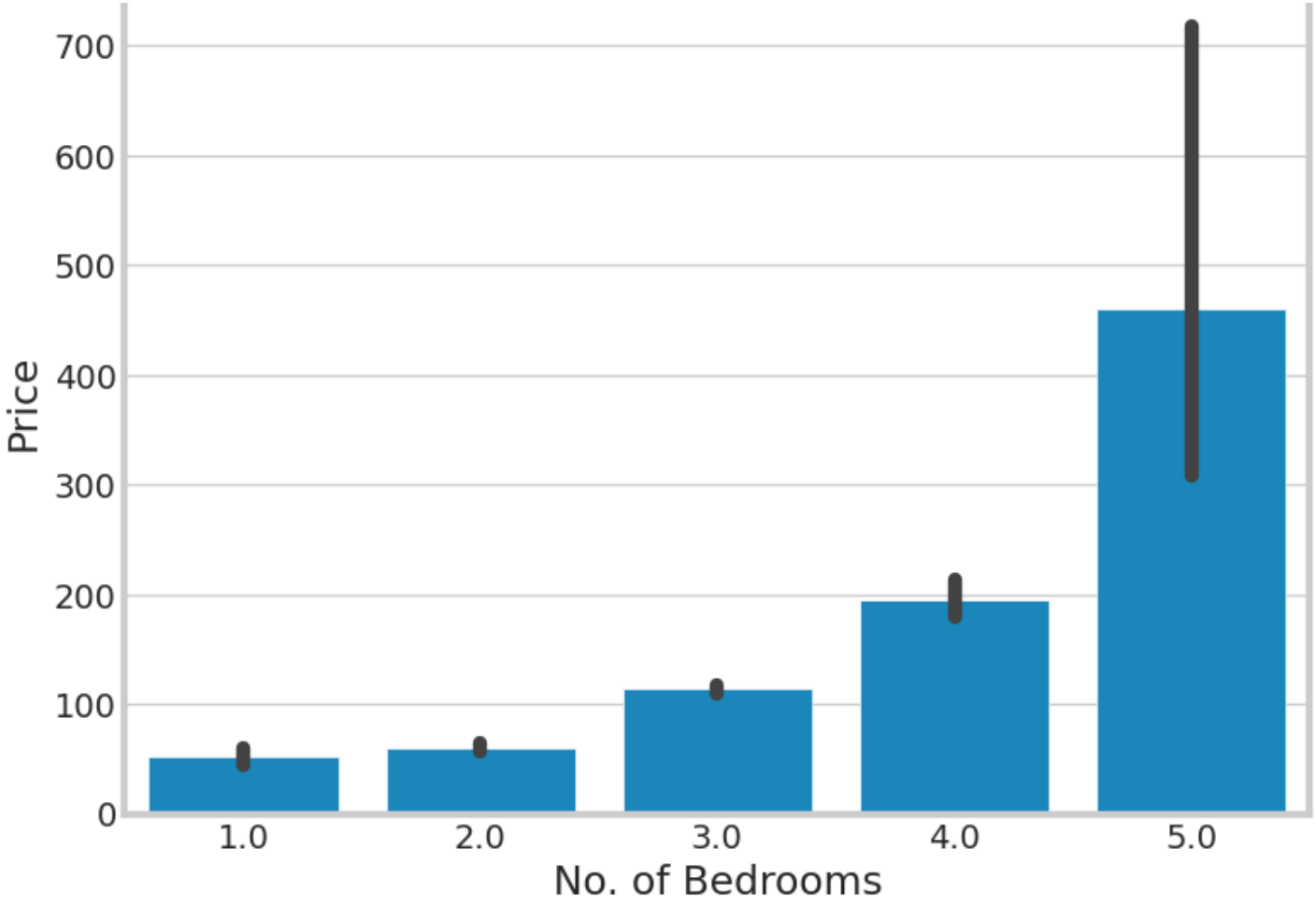
```
plt.figure(figsize=(10, 7))
sns.barplot(x='City', y='Price', hue='Resale', data=df)
plt.show()
```



Number of Bedrooms

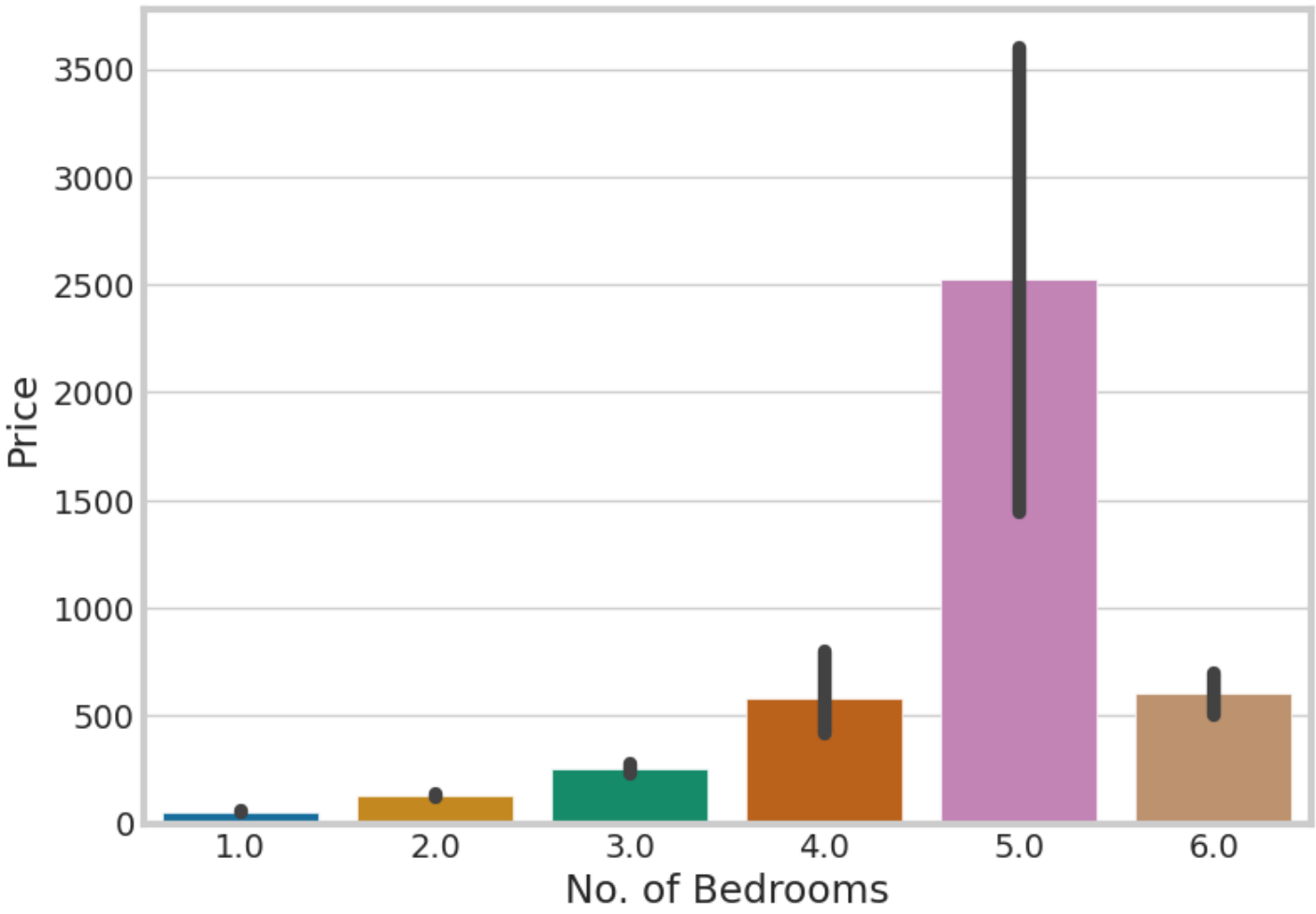
In [63]:

```
tmpdf = df[df['City'] == 'Delhi']
plt.figure(figsize=(8, 6))
sns.barplot(x='No. of Bedrooms', y='Price', data=tmpdf)
plt.show()
```



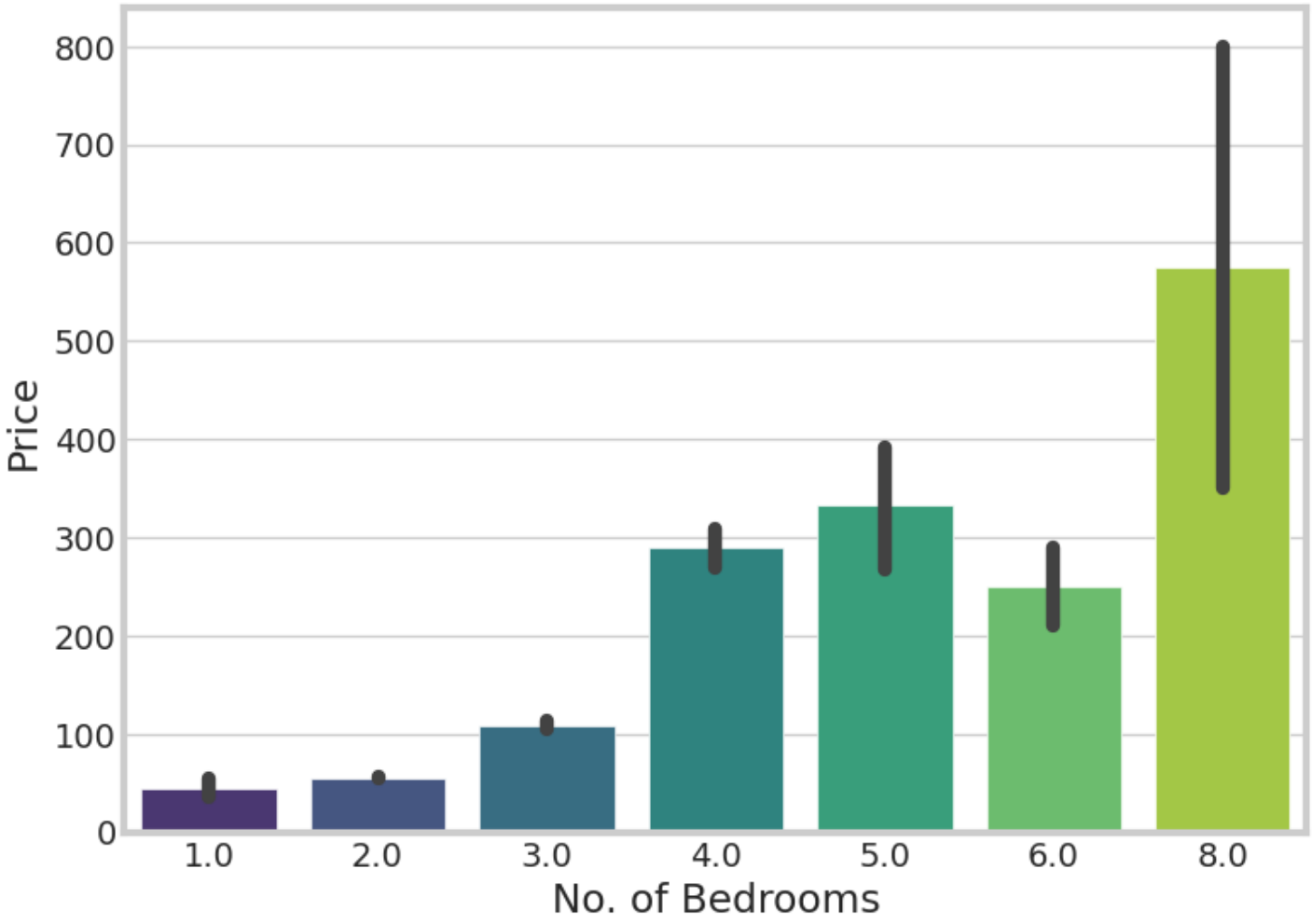
In [65]:

```
tmpdf = df[df['City'] == 'Mumbai']
plt.figure(figsize=(8, 6))
sns.barplot(x='No. of Bedrooms',palette='colorblind', y='Price', data=tmpdf)
plt.show()
```



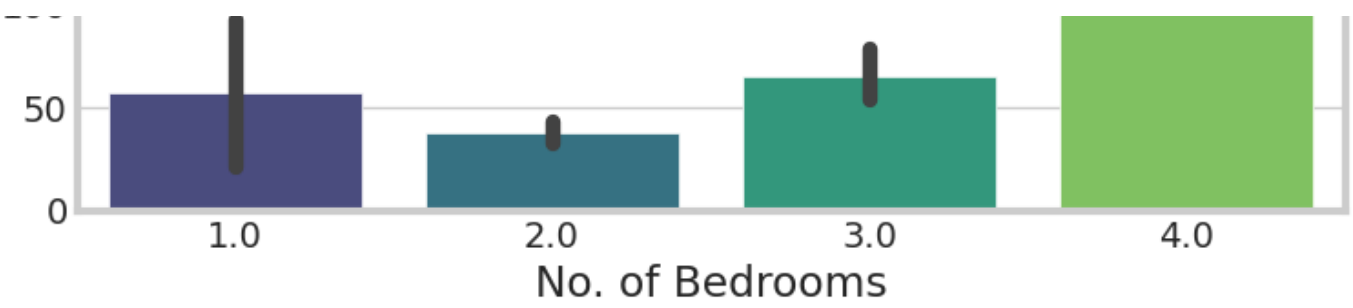
In [67]:

```
tmpdf = df[df['City'] == 'Hyderabad']
plt.figure(figsize=(8, 6))
sns.barplot(x='No. of Bedrooms',palette='viridis', y='Price', data=tmpdf)
plt.show()
```



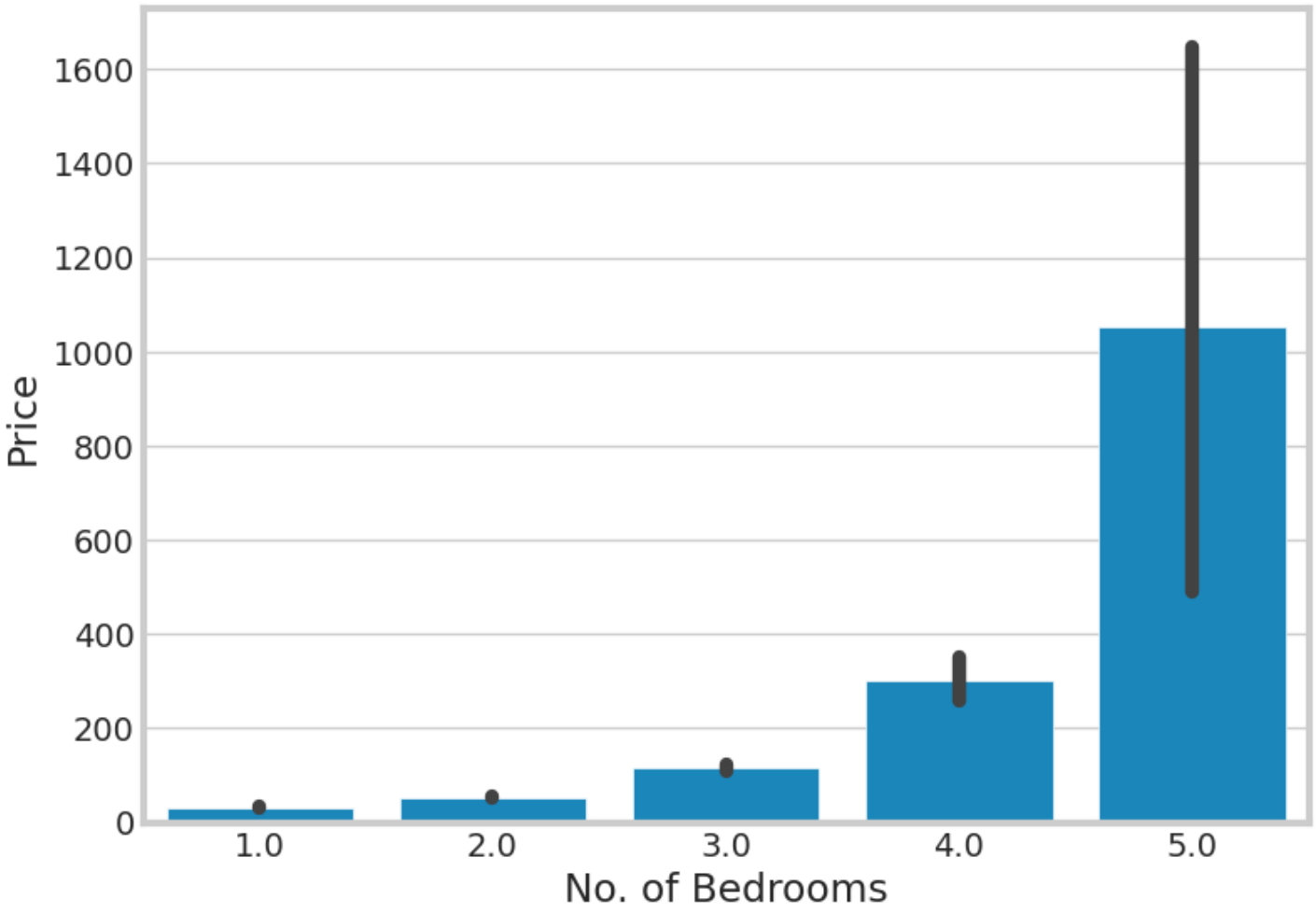
In [68]:

```
tmpdf = df[df['City'] == 'Kolkata']
plt.figure(figsize=(8, 6))
sns.barplot(x='No. of Bedrooms',palette='viridis', y='Price', data=tmpdf)
plt.show()
```



In [69]:

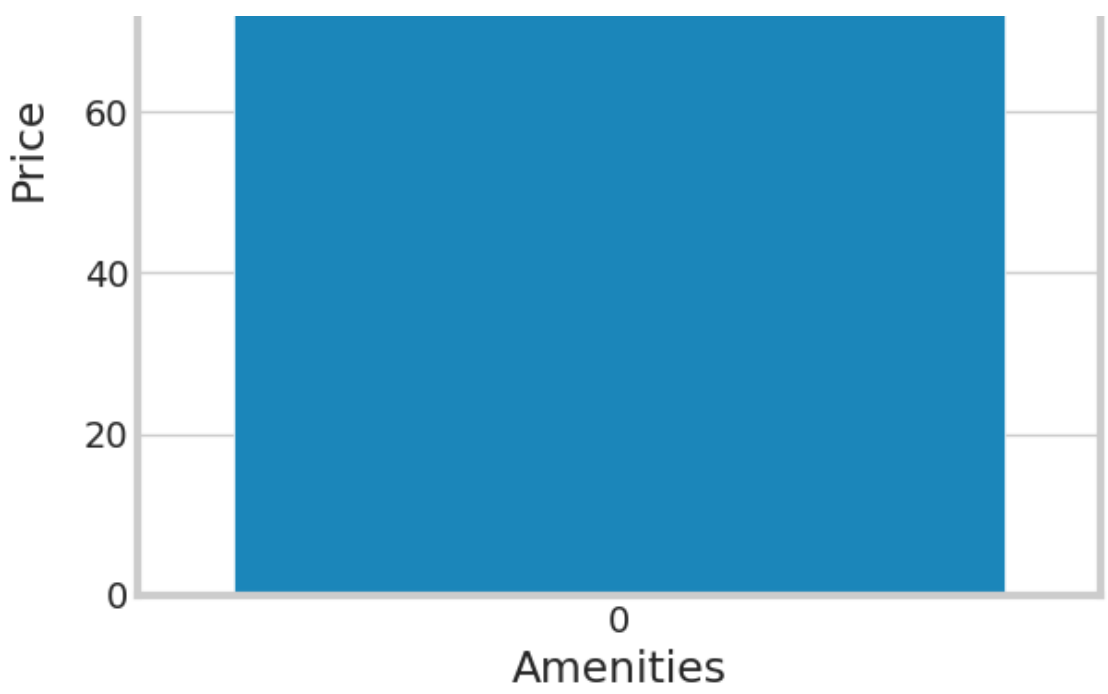
```
tmpdf = df[df['City'] == 'Chennai']
plt.figure(figsize=(8, 6))
sns.barplot(x='No. of Bedrooms', y='Price', data=tmpdf)
plt.show()
```



Amenities [Frequncy]

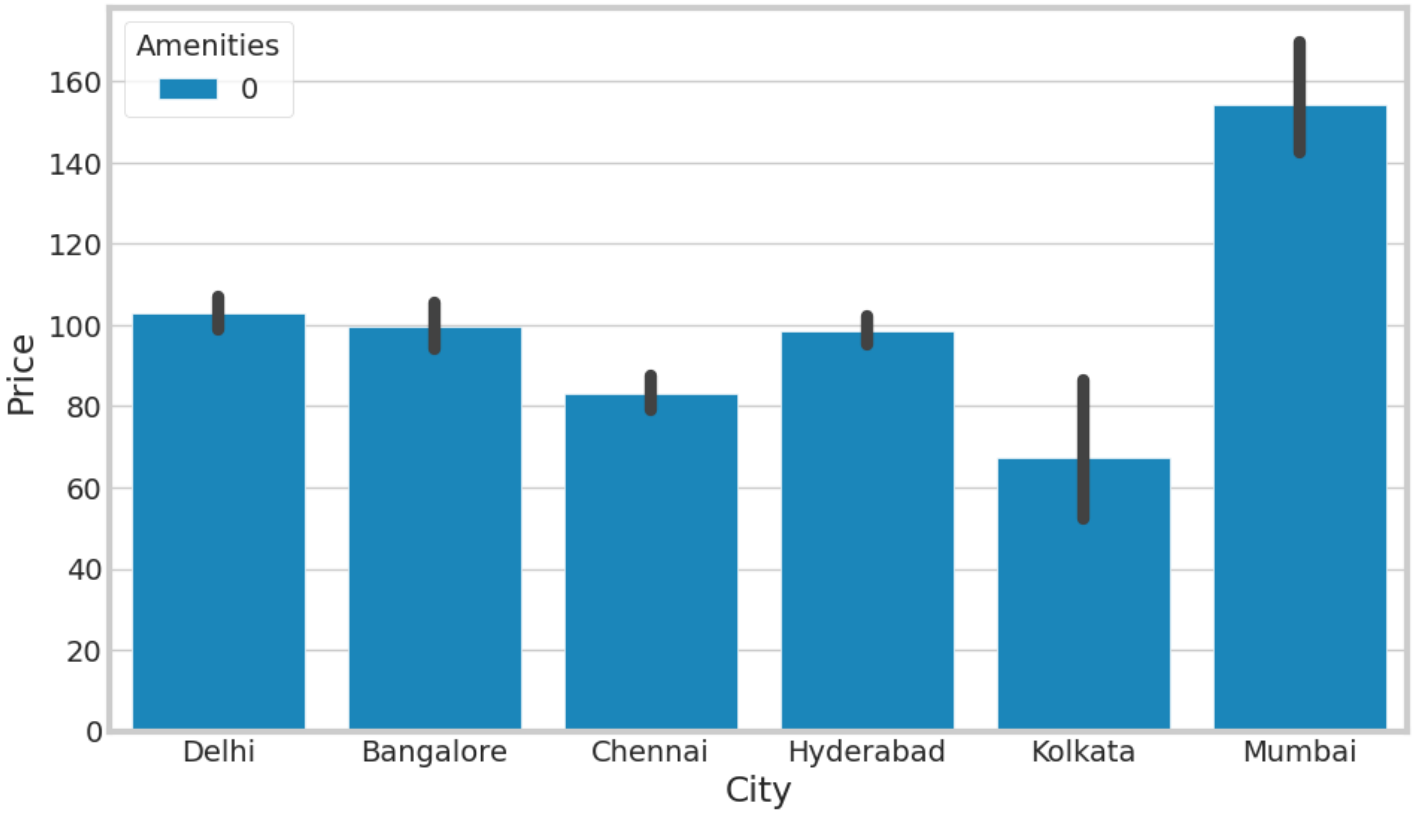
In [92]:

```
plt.figure(figsize=(6, 6))
sns.barplot(x='Amenities', y='Price', data=df)
plt.show()
```



In [83]:

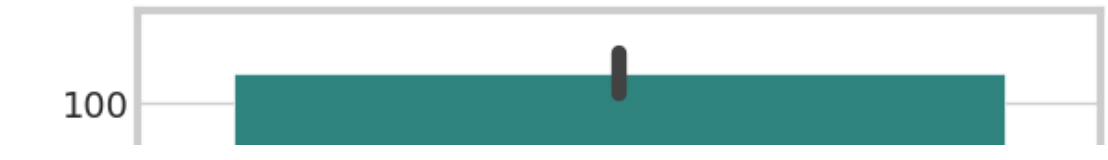
```
plt.figure(figsize=(10, 6))
sns.barplot(x='City', y='Price', hue='Amenities', data=df)
plt.show()
```

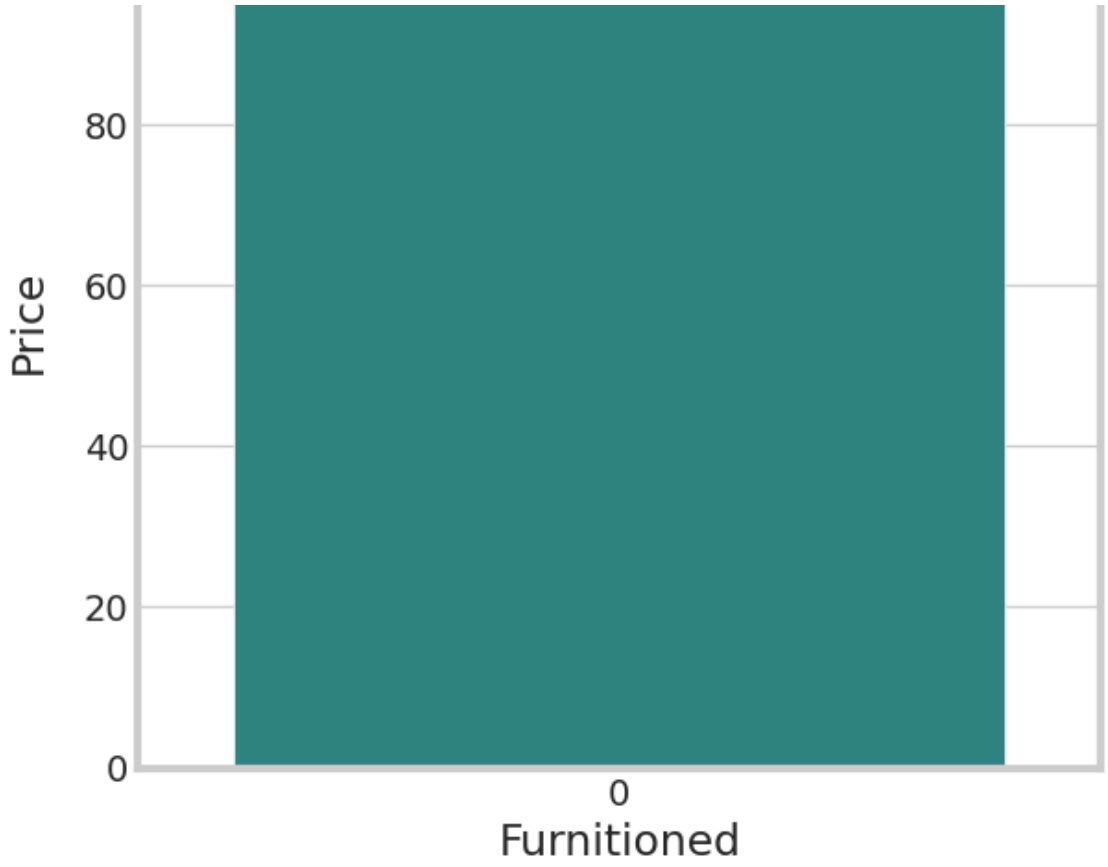


Furnitioned

In [103]:

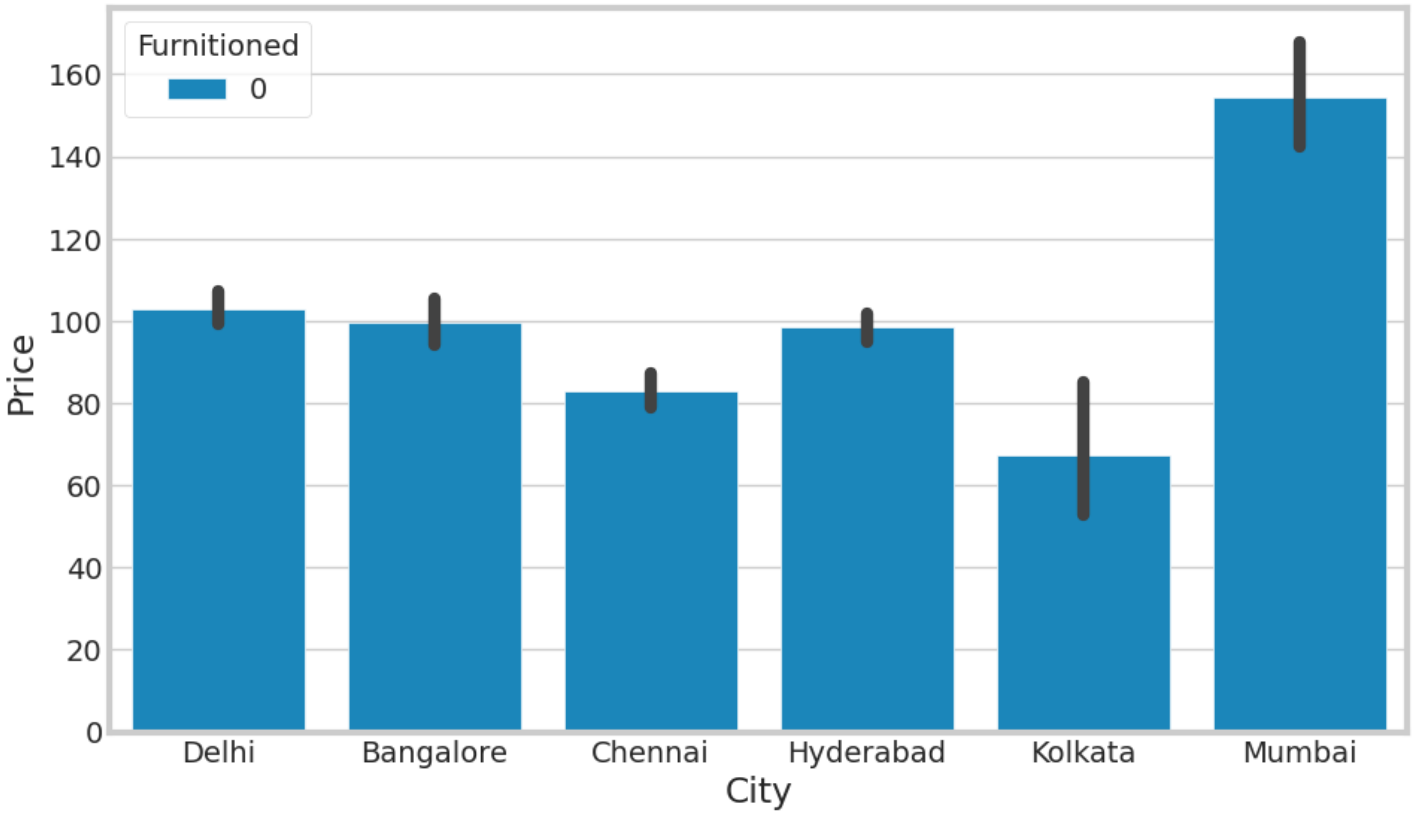
```
plt.figure(figsize=(6, 6))
sns.barplot(x=df['Furnitioned'], y=df['Price'],palette='viridis',data=df)
plt.show()
```





In [102]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['City'], y=df['Price'],hue=df['Furnitioned'])
plt.show()
```

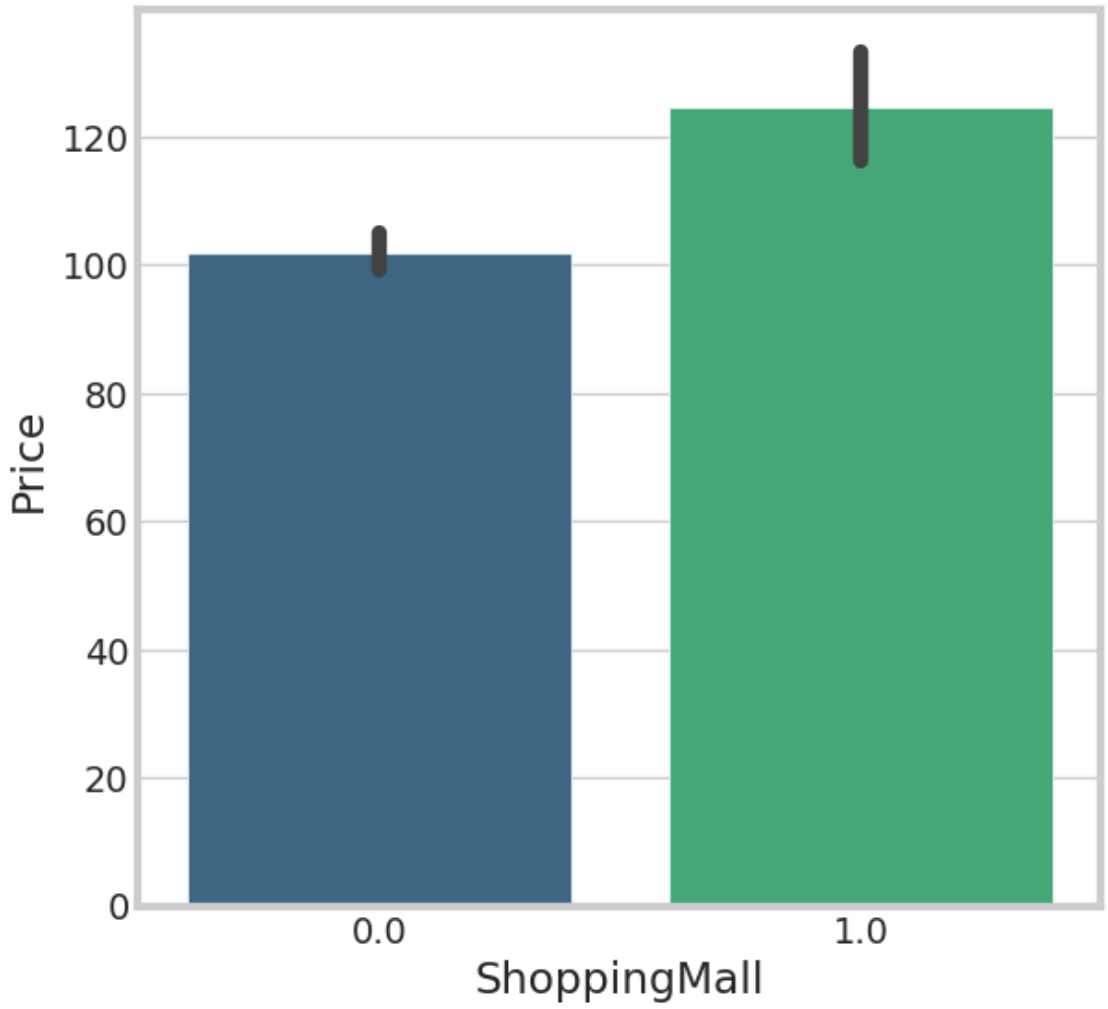


Shopping

In [95]:

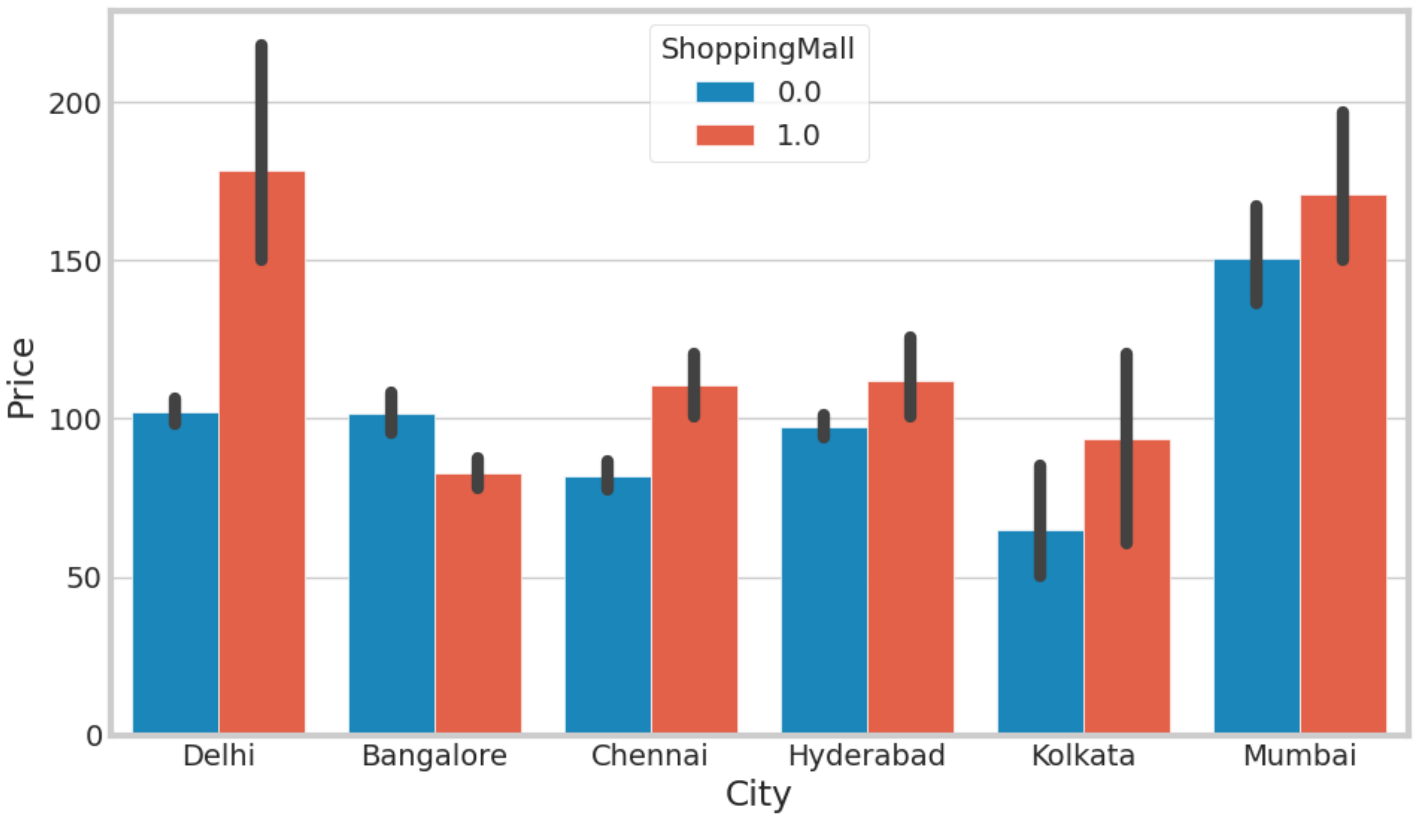
```
plt.figure(figsize=(6, 6))
sns.barplot(x=df['ShoppingMall'], y=df['Price'], palette='viridis')
```

```
plt.show()
```



In [99]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['City'], y=df['Price'],hue=df['ShoppingMall'])
plt.show()
```

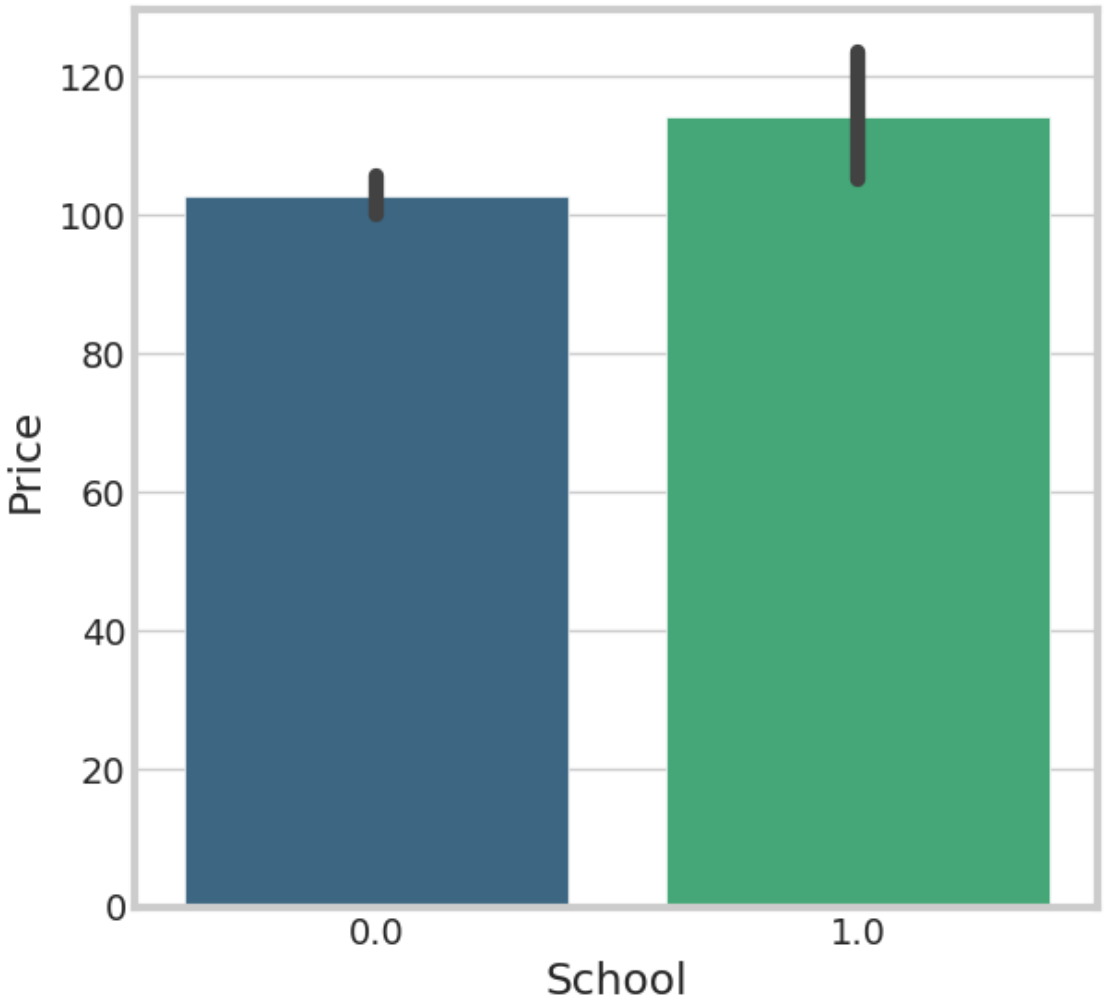


School

School

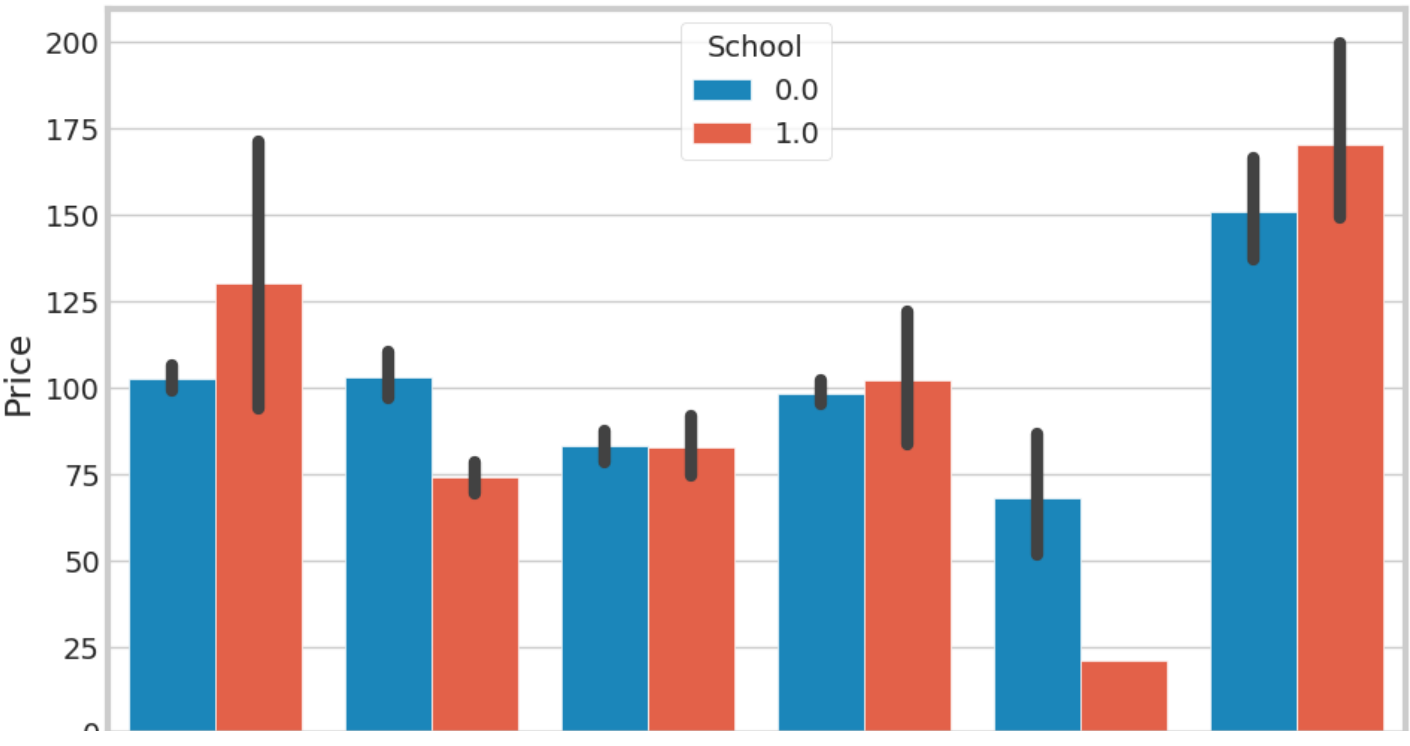
In [106]:

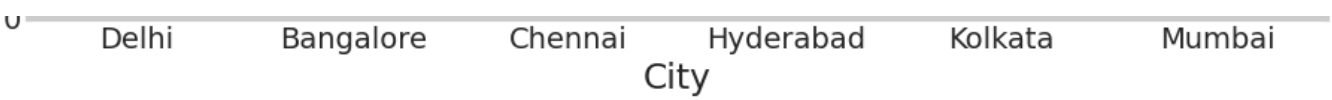
```
plt.figure(figsize=(6, 6))
sns.barplot(x=df['School'], y=df['Price'], palette='viridis')
plt.show()
```



In [108]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['City'], y=df['Price'], hue=df['School'])
plt.show()
```

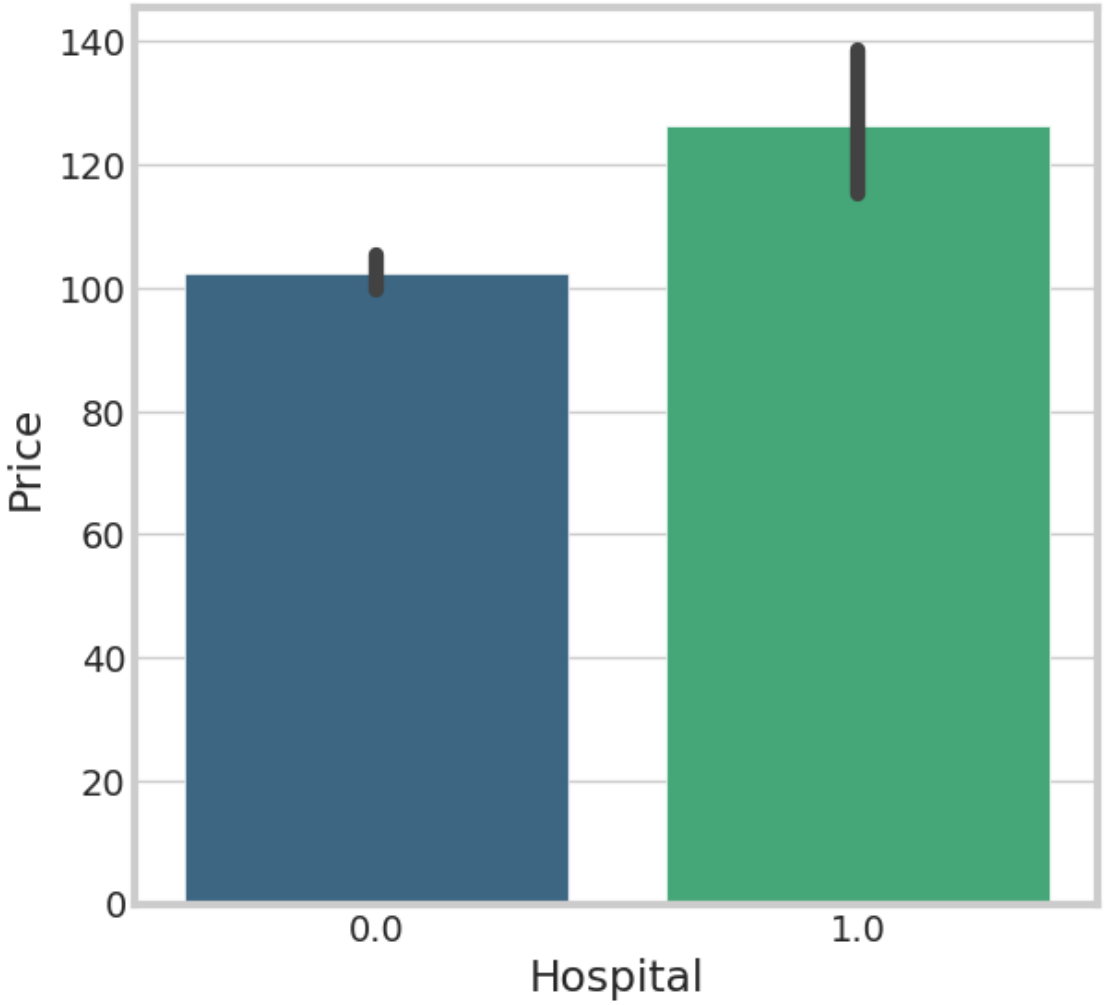




Hospital

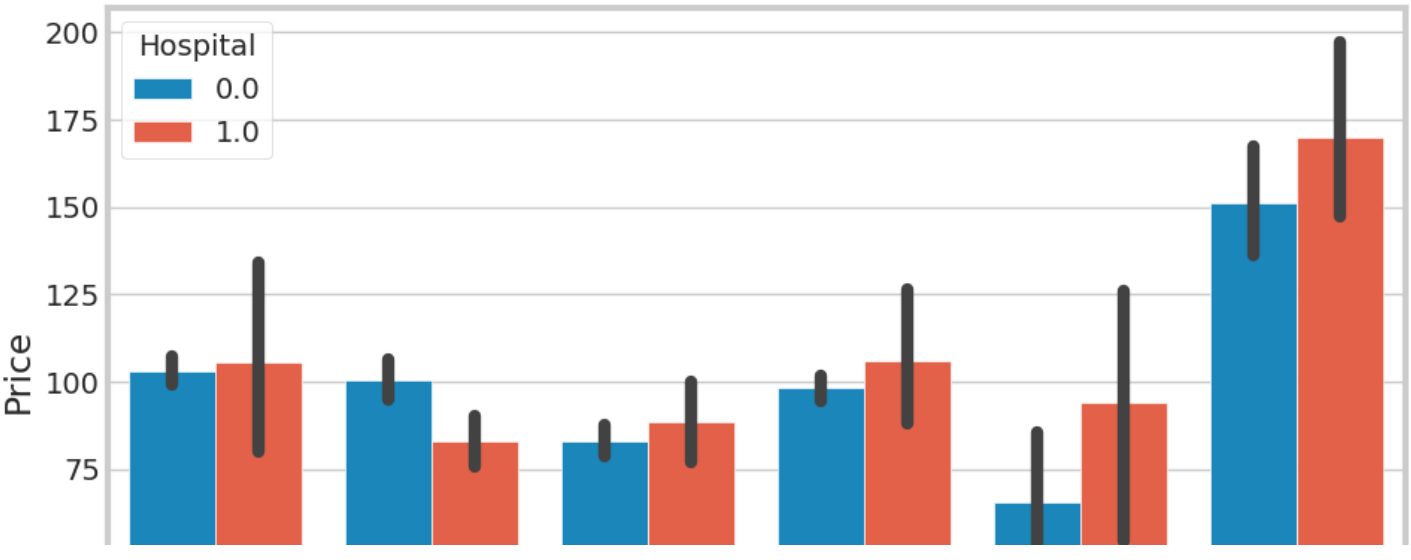
In [110]:

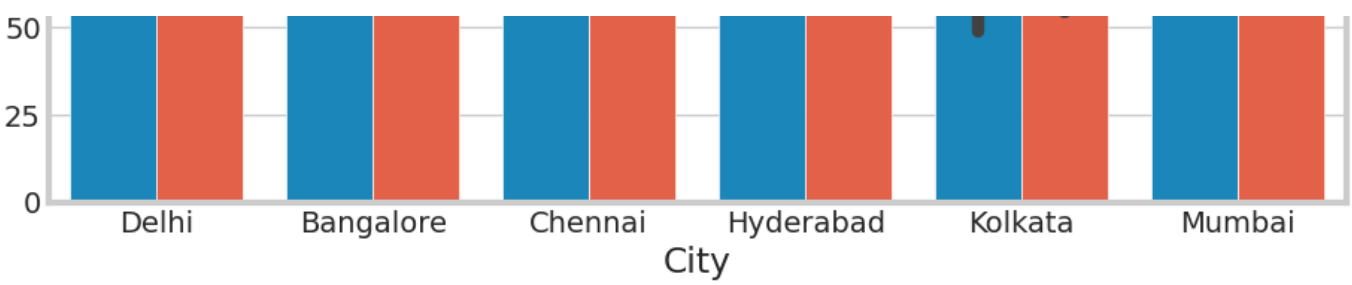
```
plt.figure(figsize=(6, 6))
sns.barplot(x=df['Hospital'], y=df['Price'],palette='viridis')
plt.show()
```



In [111]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['City'], y=df['Price'], hue=df['Hospital'])
plt.show()
```





Data Preprocessing

In [112]:

```
savedf = df.copy()
tmpdf = df.drop(columns=['Location', 'City'], axis=1)
```

In [113]:

```
from sklearn.preprocessing import StandardScaler

scalar = StandardScaler()

tmpdf = pd.DataFrame(columns=tmpdf.columns, data=scalar.fit_transform(tmpdf))
```

In [114]:

```
tmpdf.head()
```

Out[114]:

	Unnamed: 0	Price	Area	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	Amenit
0	-1.481660	0.010078	-0.210550	-0.605067	1.385691	-0.28375	-0.335494	-0.280886	-0.245574	1.534982	
1	-1.481537	-0.320785	-0.493093	0.711237	-0.721662	-0.28375	-0.335494	-0.280886	-0.245574	-0.651474	
2	-1.481413	0.340941	0.001357	-0.605067	1.385691	-0.28375	-0.335494	-0.280886	-0.245574	-0.651474	
3	-1.481290	-0.578123	-1.291277	-0.605067	-0.721662	-0.28375	-0.335494	-0.280886	-0.245574	1.534982	
4	-1.481167	-0.335490	-0.634365	0.711237	-0.721662	-0.28375	-0.335494	-0.280886	-0.245574	-0.651474	

DIMENSIONALITY REDUCTION

In [115]:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=3)
x = pca.fit_transform(tmpdf)
pca.explained_variance_ratio_ # percentage of variance captured by each principal compone
nt
```

Out[115]:

```
array([0.29497837, 0.27569914, 0.10419294])
```

In [116]:

```
pca_df = pd.DataFrame(data=x, columns=['col1', 'col2', 'col3'])
```

In [117]:

```
pca_df.head()
```

Out[117]:

	col1	col2	col3
0	-0.446441	-0.320290	0.147338
1	-0.763731	0.807736	-1.756476
2	-0.958224	-0.039629	0.157428
3	-0.823817	-1.055904	-1.373461
4	-0.796080	0.737261	-1.780822

In [118]:

```
pca_df.describe().T
```

Out[118]:

	count	mean	std	min	25%	50%	75%	max
col1	10092.0	6.759027e-17	1.801412	-1.923210	-1.013852	-0.488637	0.209127	10.481496
col2	10092.0	9.012036e-17	1.741548	-5.620632	-0.886667	-0.482840	1.012599	15.025341
col3	10092.0	4.506018e-17	1.070624	-2.566104	-0.639021	-0.182498	0.287283	13.450132

CLUSTERING

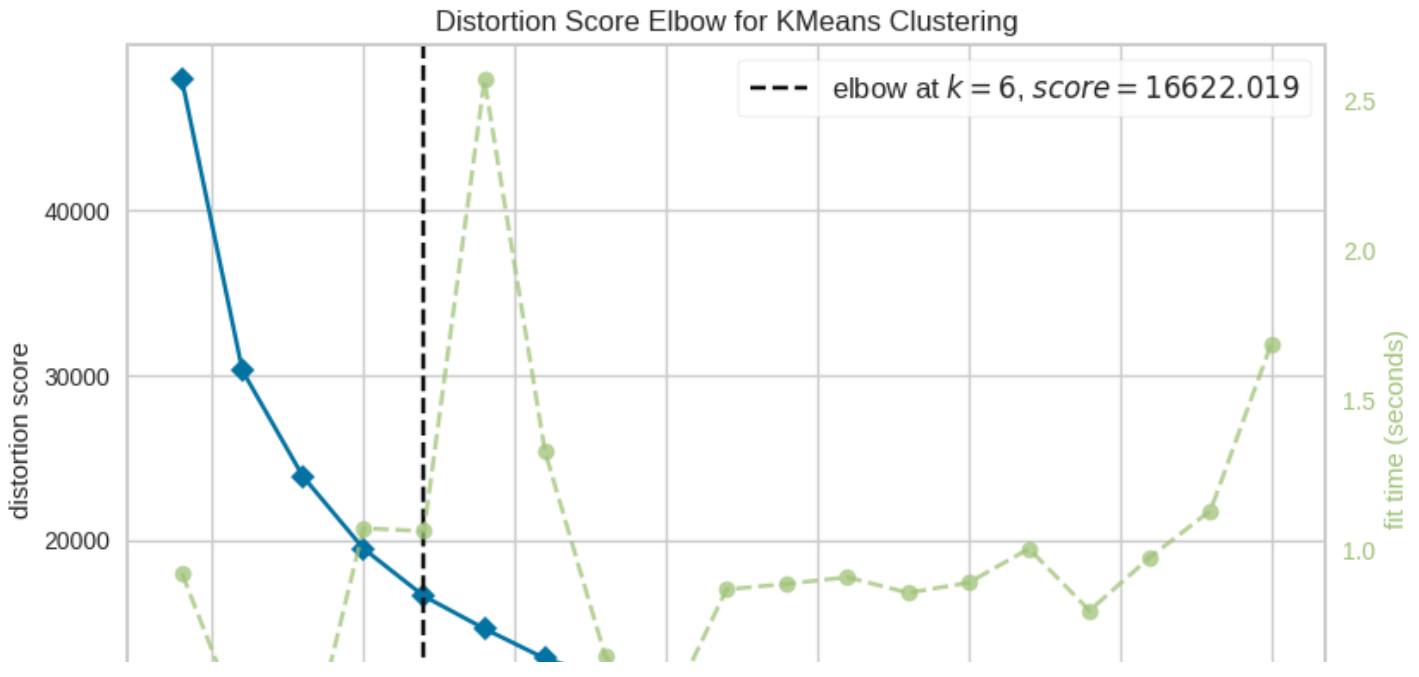
In [119]:

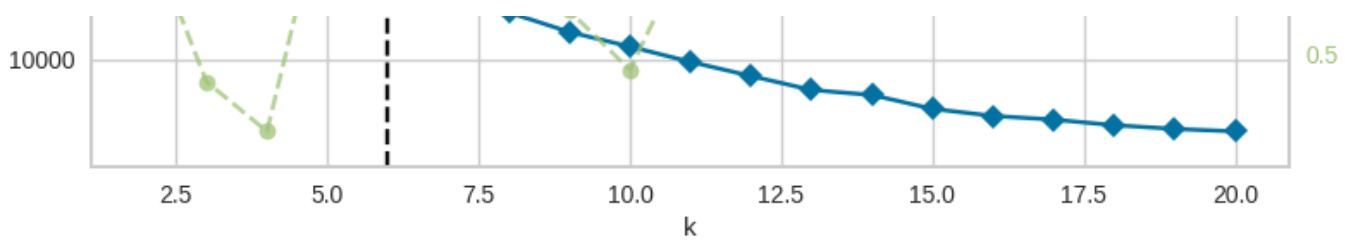
```
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
```

In [120]:

```
# Quick examination of elbow method to find numbers of clusters to make.
print('Elbow Method to determine the number of clusters to be formed:')
Elbow_M = KElbowVisualizer(KMeans(), k=20)
Elbow_M.fit(pca_df)
Elbow_M.show()
```

```
Elbow Method to determine the number of clusters to be formed:
```





Out[120]:

```
<Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>
```

According to elbow curve, optimal no. of clusters should be 7, but for simplicity we will take 5 clusters

In [121]:

```
cluster_obj = AgglomerativeClustering(n_clusters=5)
y_pred = cluster_obj.fit_predict(pca_df)
df['clusters'] = y_pred
pca_df['clusters'] = y_pred
```

In [122]:

```
df.head()
```

Out[122]:

	Unnamed: 0	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amen
0	0	105.0	1200	Sector 10 Dwarka	2.0	1	0.0	0.0	0.0	0.0	1.0	Delhi	
1	1	60.0	1000	Uttam Nagar	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	
2	2	150.0	1350	Sarita Vihar	2.0	1	0.0	0.0	0.0	0.0	0.0	Delhi	
3	3	25.0	435	Uttam Nagar	2.0	0	0.0	0.0	0.0	0.0	1.0	Delhi	
4	4	58.0	900	Dwarka Mor	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	

Segment profiling

In [123]:

```
df = pd.read_csv('clustred_data.csv')
```

In [124]:

```
df.drop(columns=['Unnamed: 0'], axis=1, inplace=True)
df.head()
```

Out[124]:

	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amenities	Furnit
0	105.0	1200	Sector 10 Dwarka	2.0	1	0.0	0.0	0.0	0.0	1.0	Delhi	1	
1	60.0	1000	Uttam Nagar	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	1	

	Price	Area	Location	No. of Bedrooms	Resale	ShoppingMall	ATM	School	Hospital	VaastuCompliant	City	Amenities	Furnit
2	150.0	1350	Sarita Vihar	2.0	1	0.0	0.0	0.0	0.0	0.0	Delhi	0	
3	25.0	435	Uttam Nagar	2.0	0	0.0	0.0	0.0	0.0	1.0	Delhi	1	
4	58.0	900	Dwarka Mor	3.0	0	0.0	0.0	0.0	0.0	0.0	Delhi	1	

In [125]:

```
df['clusters'].value_counts().sort_values(ascending=False)
```

Out[125]:

```
clusters
0    4793
1    2549
2    1877
3     515
4     358
Name: count, dtype: int64
```

In [126]:

```
city = df['City'].unique()
```

Price

In [127]:

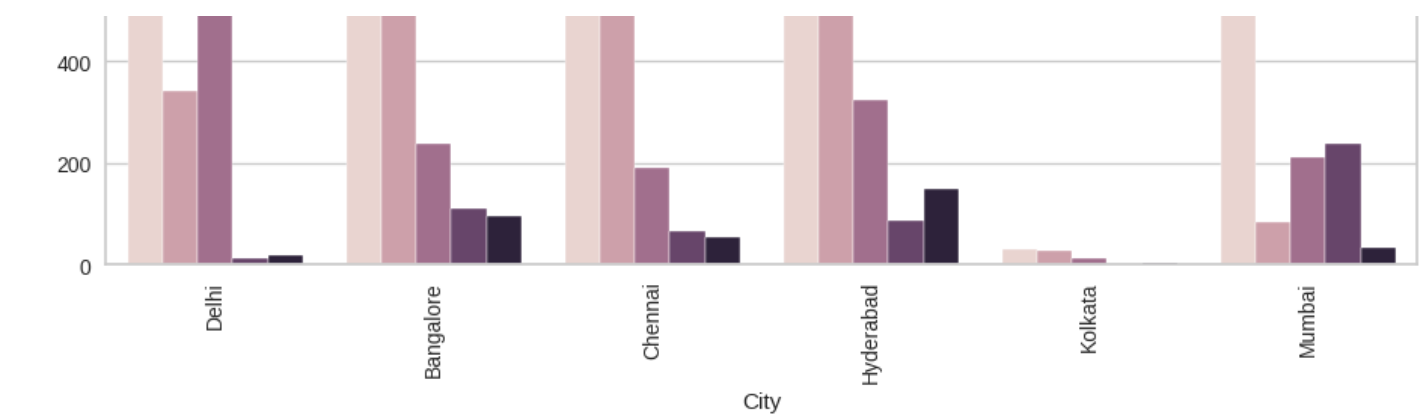
```
for i in range(0, 5):
    print(f"the average price of obeservation in {i+1} cluster is = ", df[df['clusters']
== i]['Price'].mean())
```

```
the average price of obeservation in 1 cluster is =  58.6557810682245
the average price of obeservation in 2 cluster is =  90.02948166339742
the average price of obeservation in 3 cluster is =  162.40752242408098
the average price of obeservation in 4 cluster is =  131.13586372815533
the average price of obeservation in 5 cluster is =  454.83519513966473
```

City

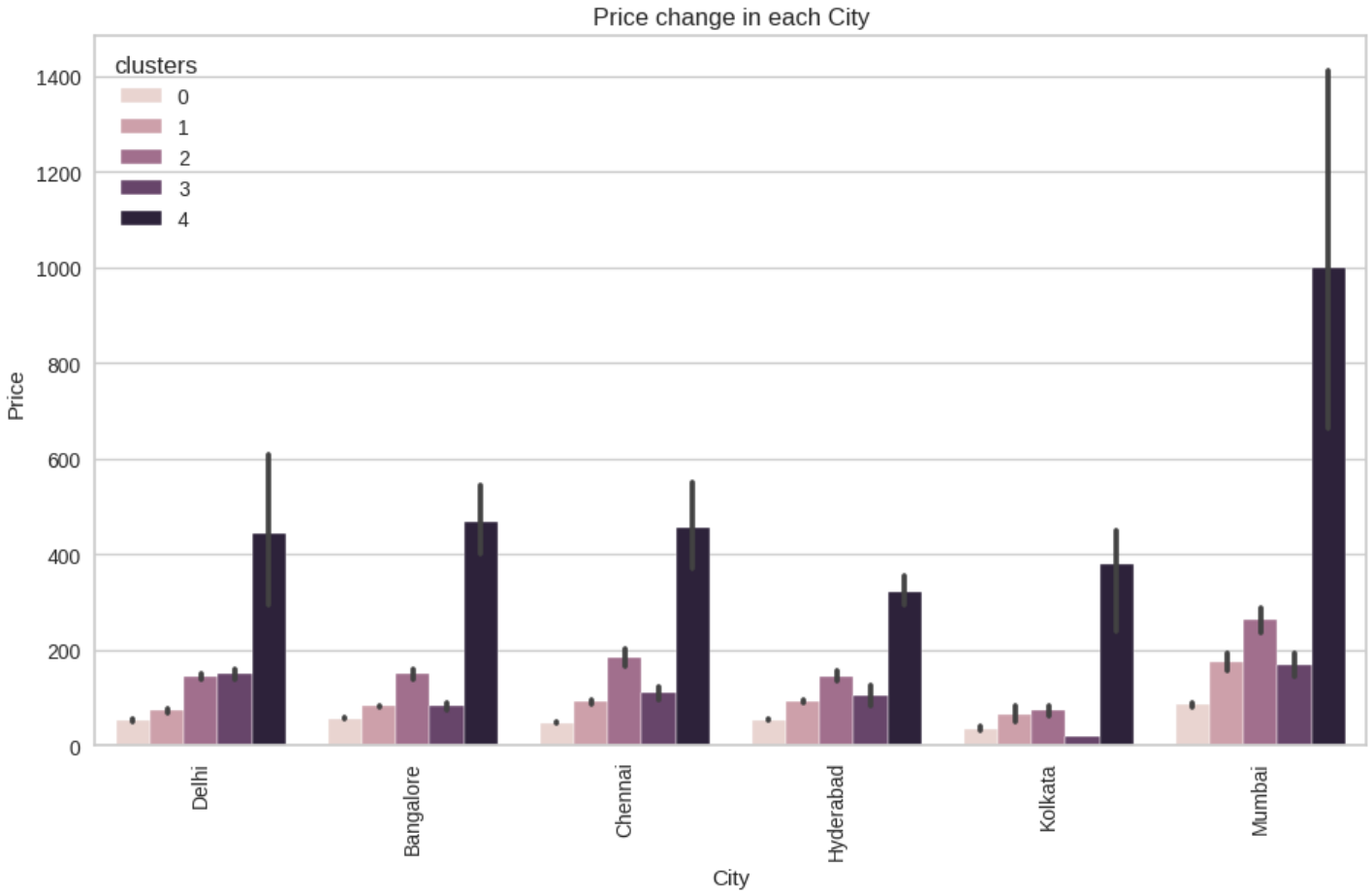
In [130]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='City', hue='clusters', data=df)
plt.title('distribution of each City cluster wise')
plt.xticks(rotation=90)
plt.show()
```



In [132]:

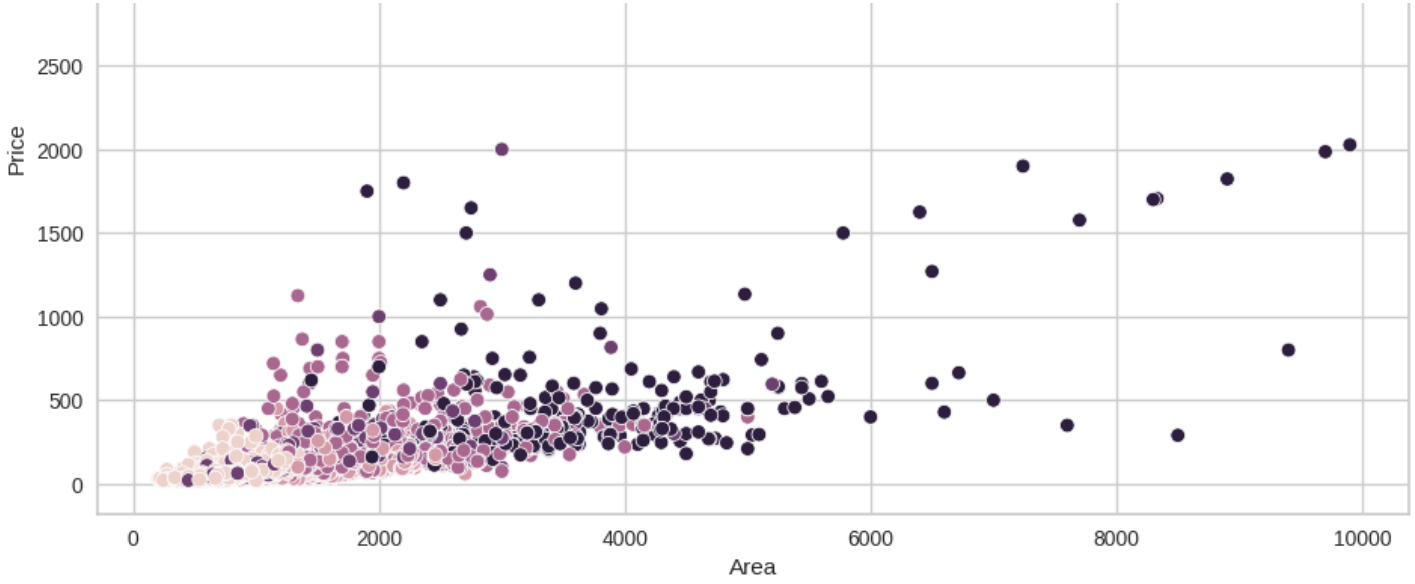
```
plt.figure(figsize=(10, 6))
sns.barplot(x='City', y='Price', hue='clusters', data=df)  # Use 'x' and 'y' to specify the variables
plt.title('Price change in each City')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.show()
```



Area

In [133]:

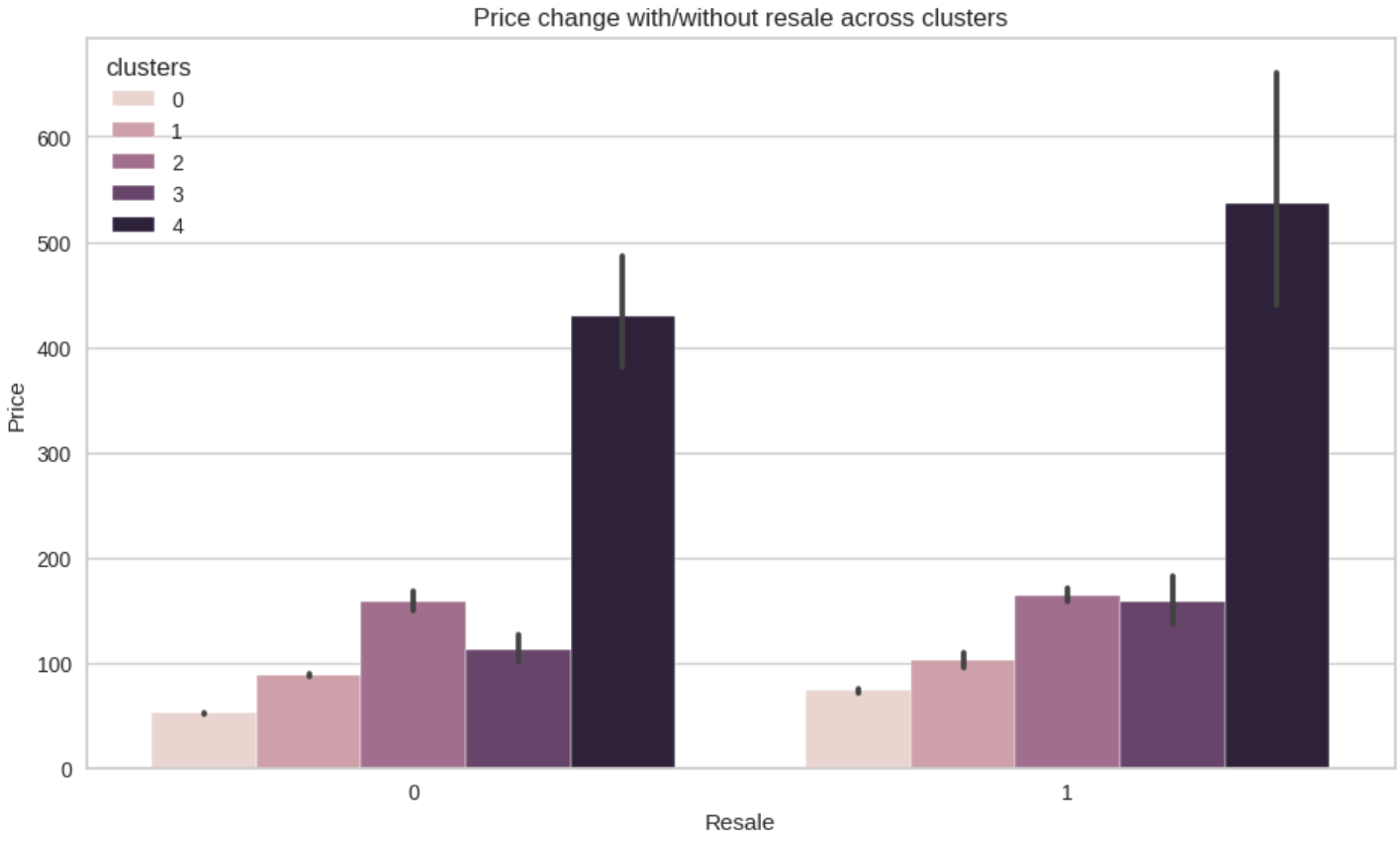
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['Area'], y=df['Price'], hue=df['clusters'])
plt.show()
```



Resale

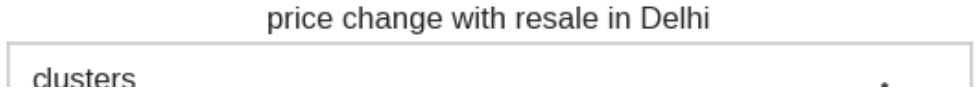
In [135]:

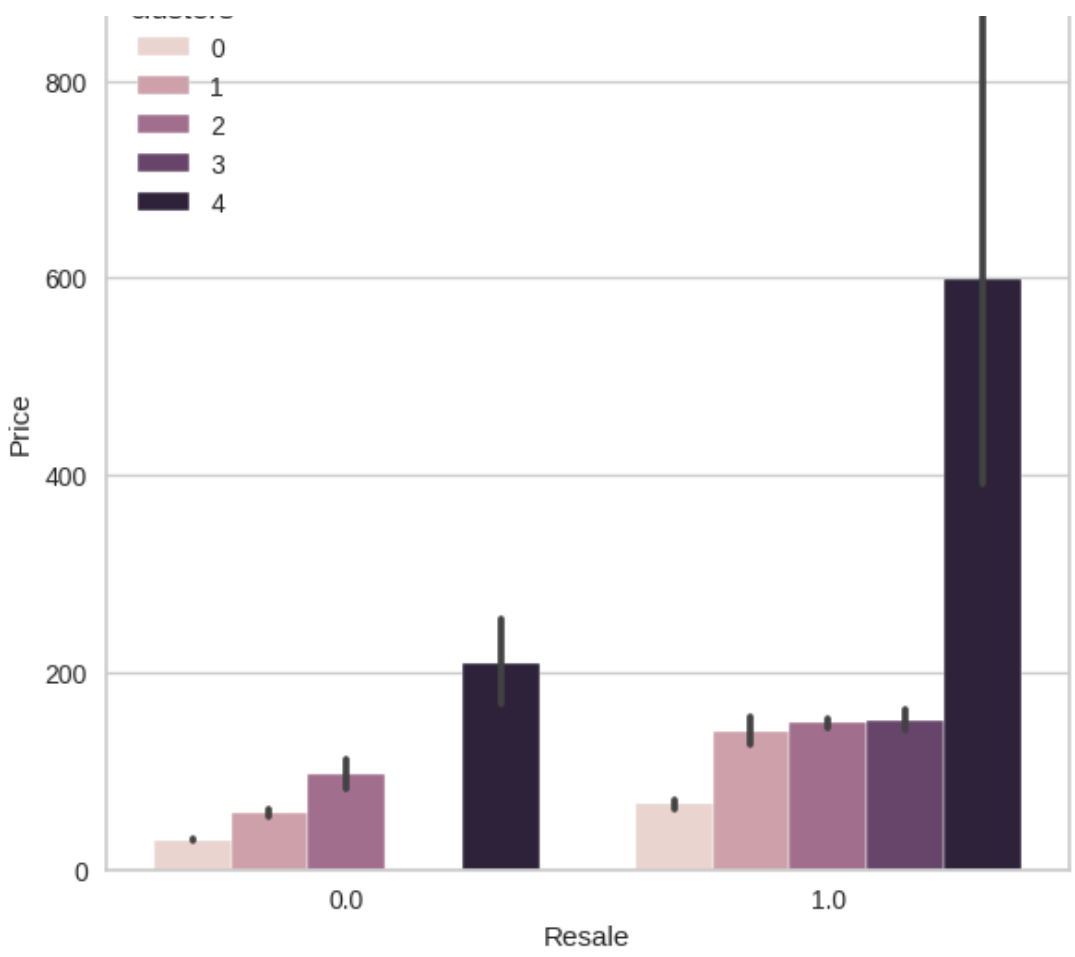
```
plt.figure(figsize=(10, 6))
sns.barplot(x='Resale', y='Price', hue='clusters', data=df)
plt.title('Price change with/without resale across clusters')
plt.show()
```



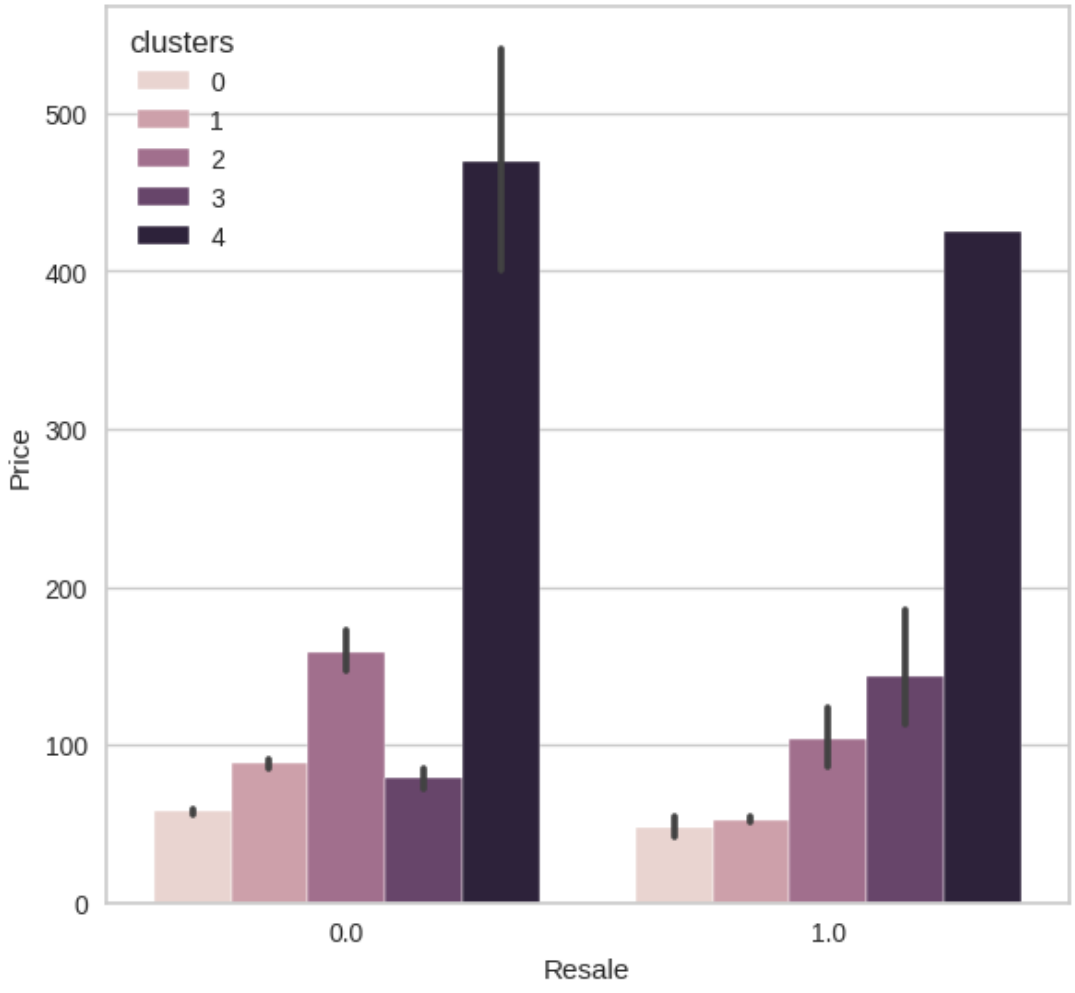
In [136]:

```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='Resale', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with resale in {i}")
  plt.show()
```

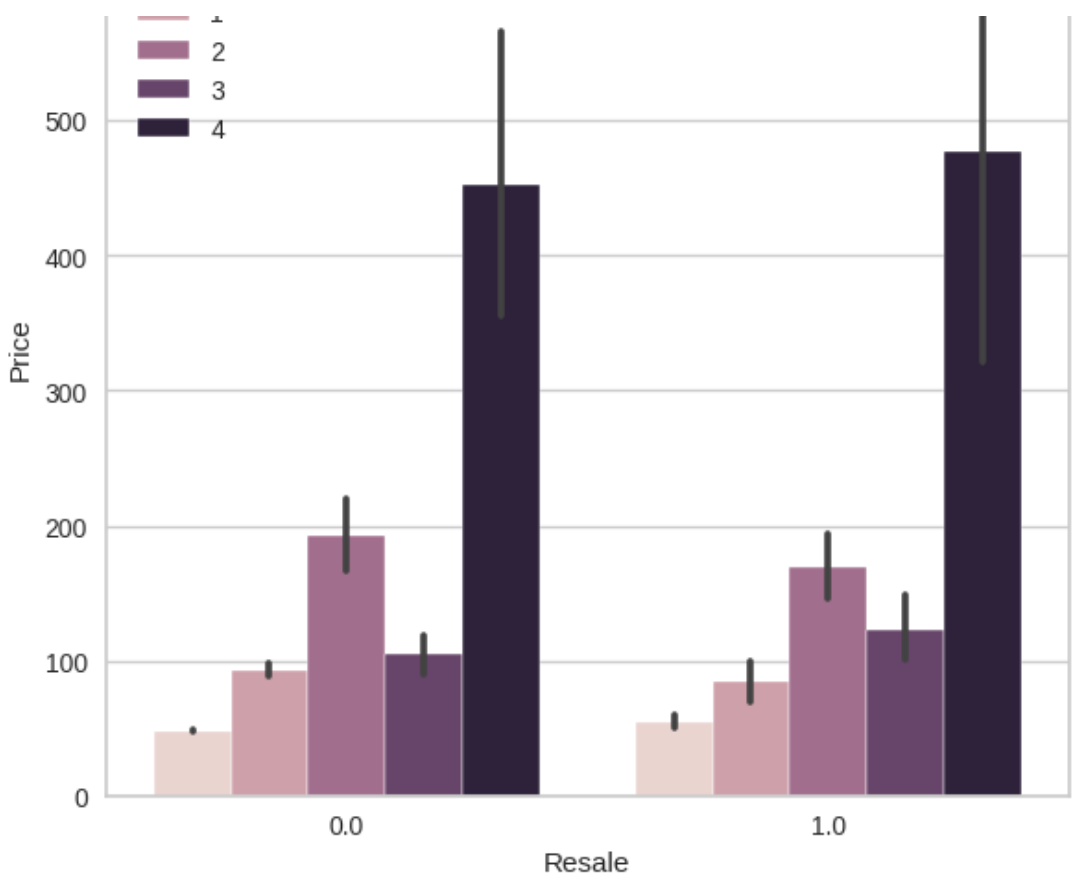




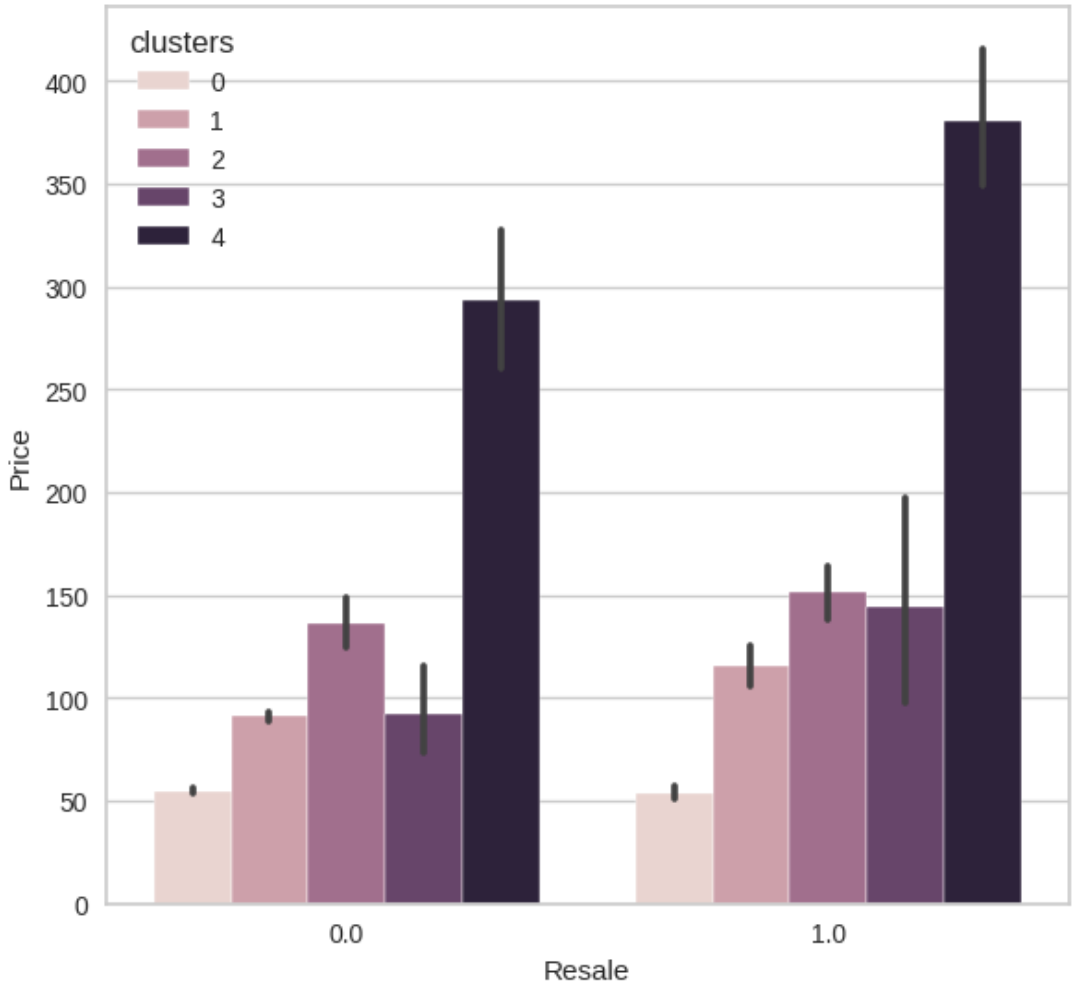
price change with resale in Bangalore



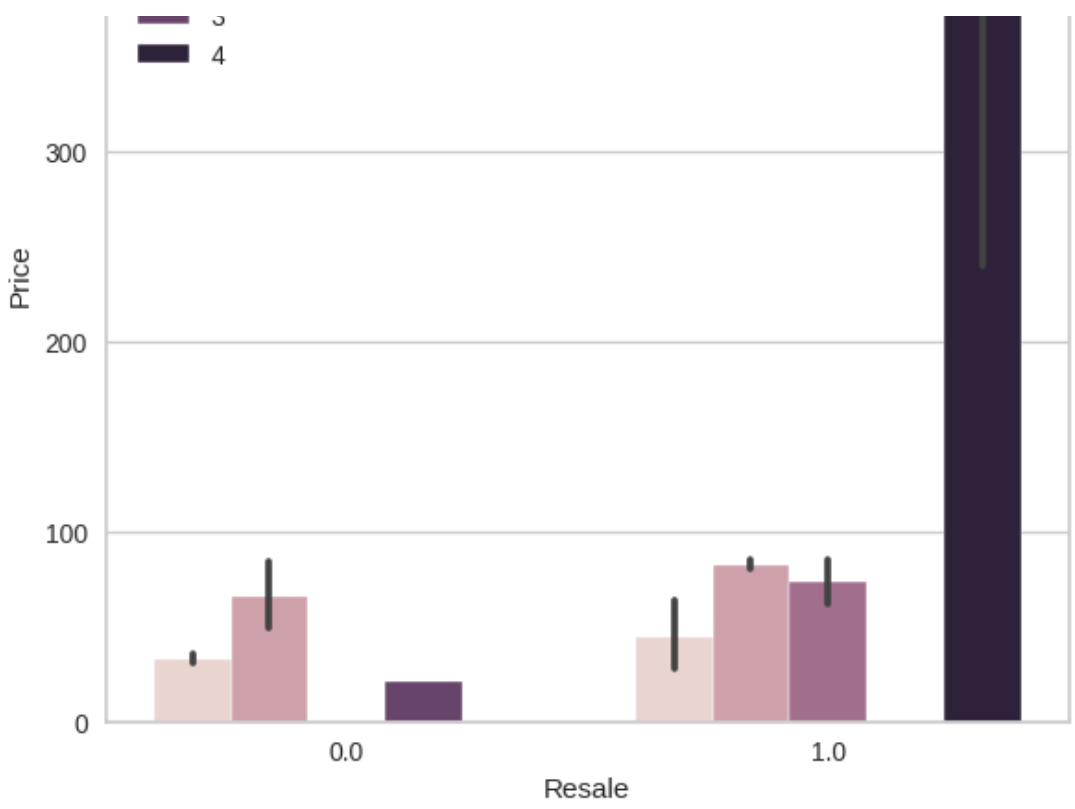
price change with resale in Chennai

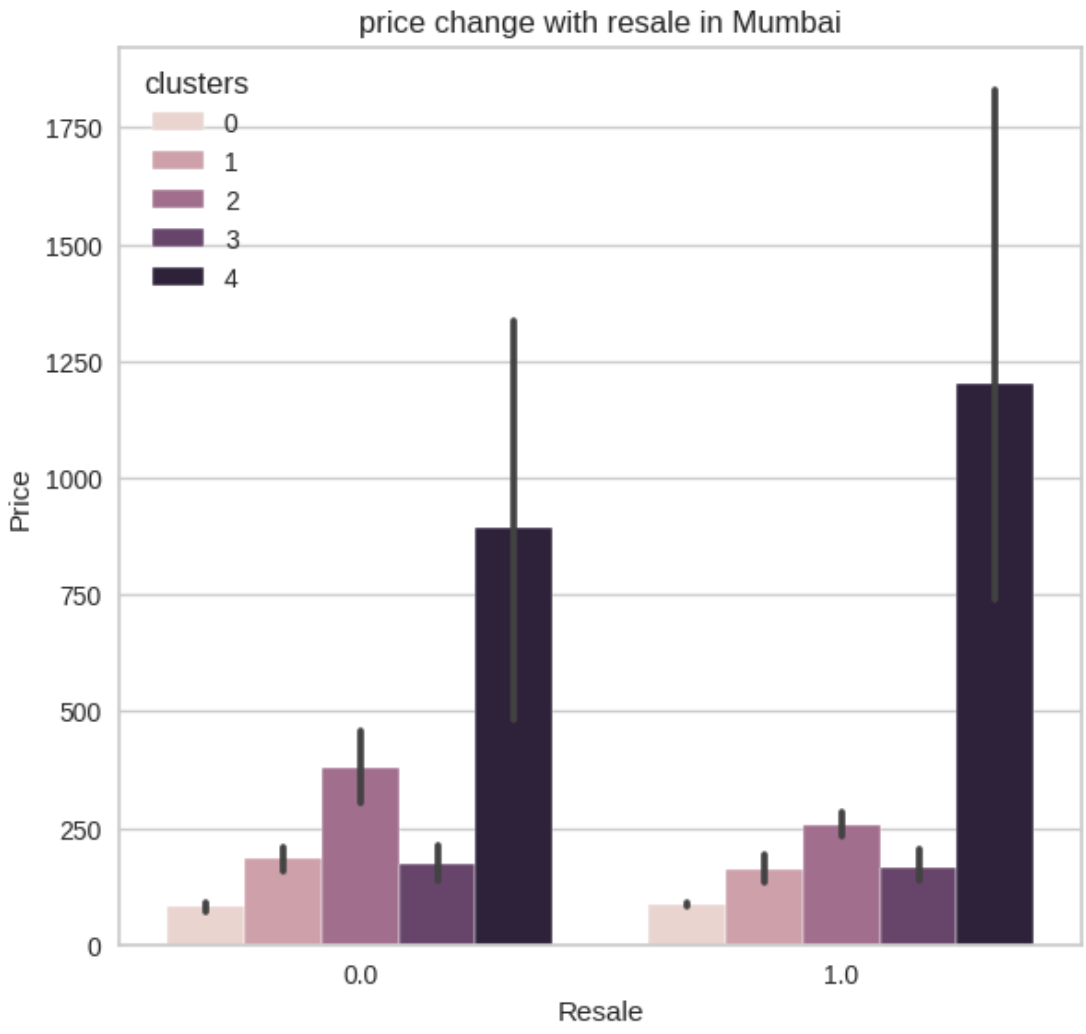


price change with resale in Hyderabad



price change with resale in Kolkata



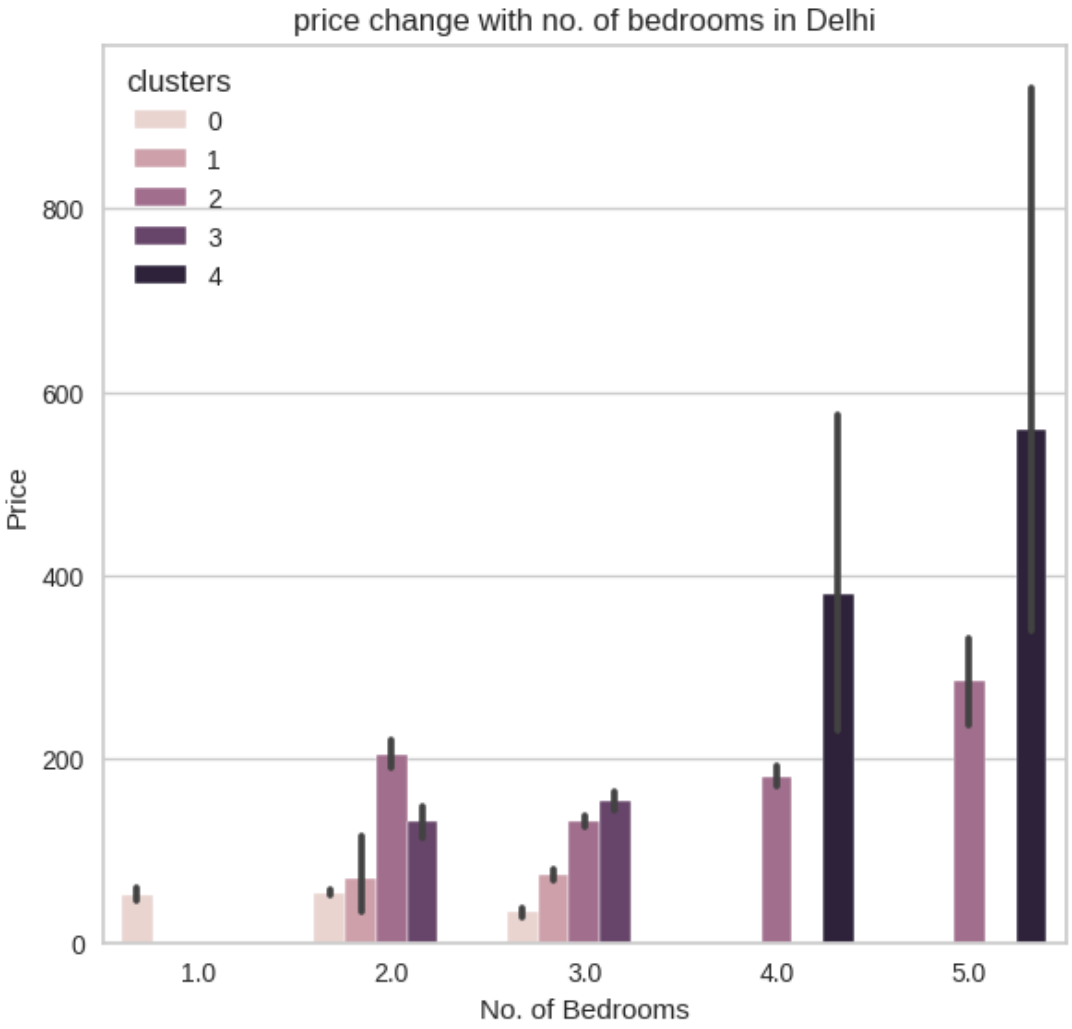


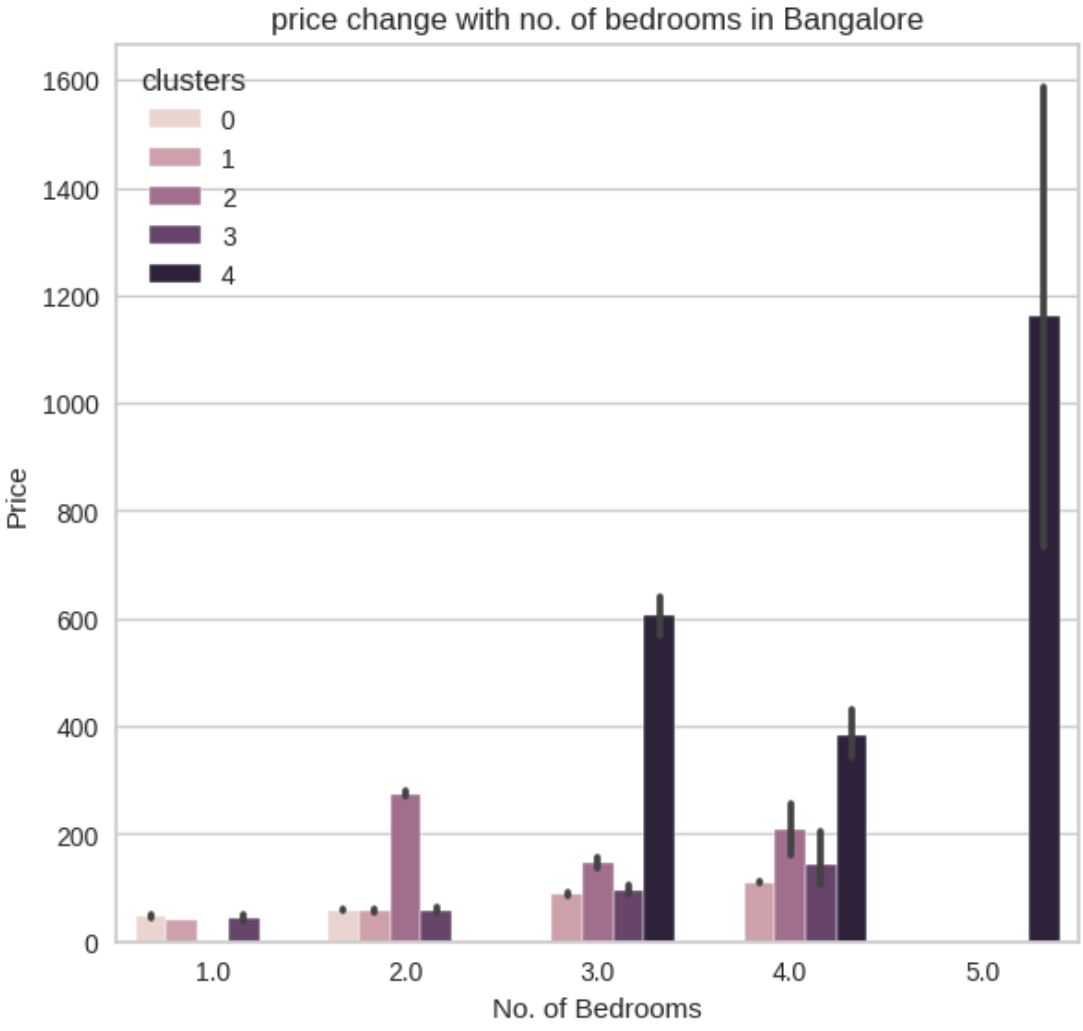
Number of bedrooms

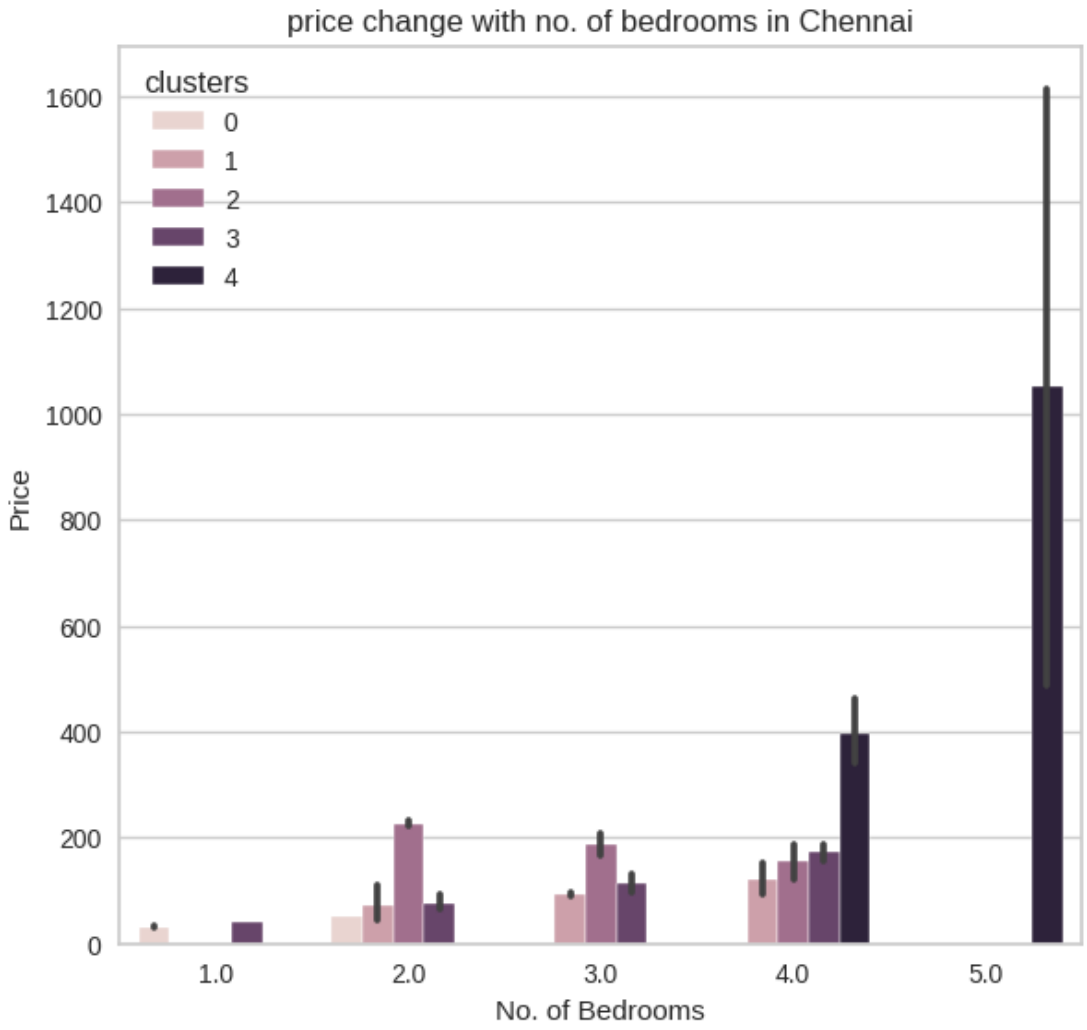
In [137]:

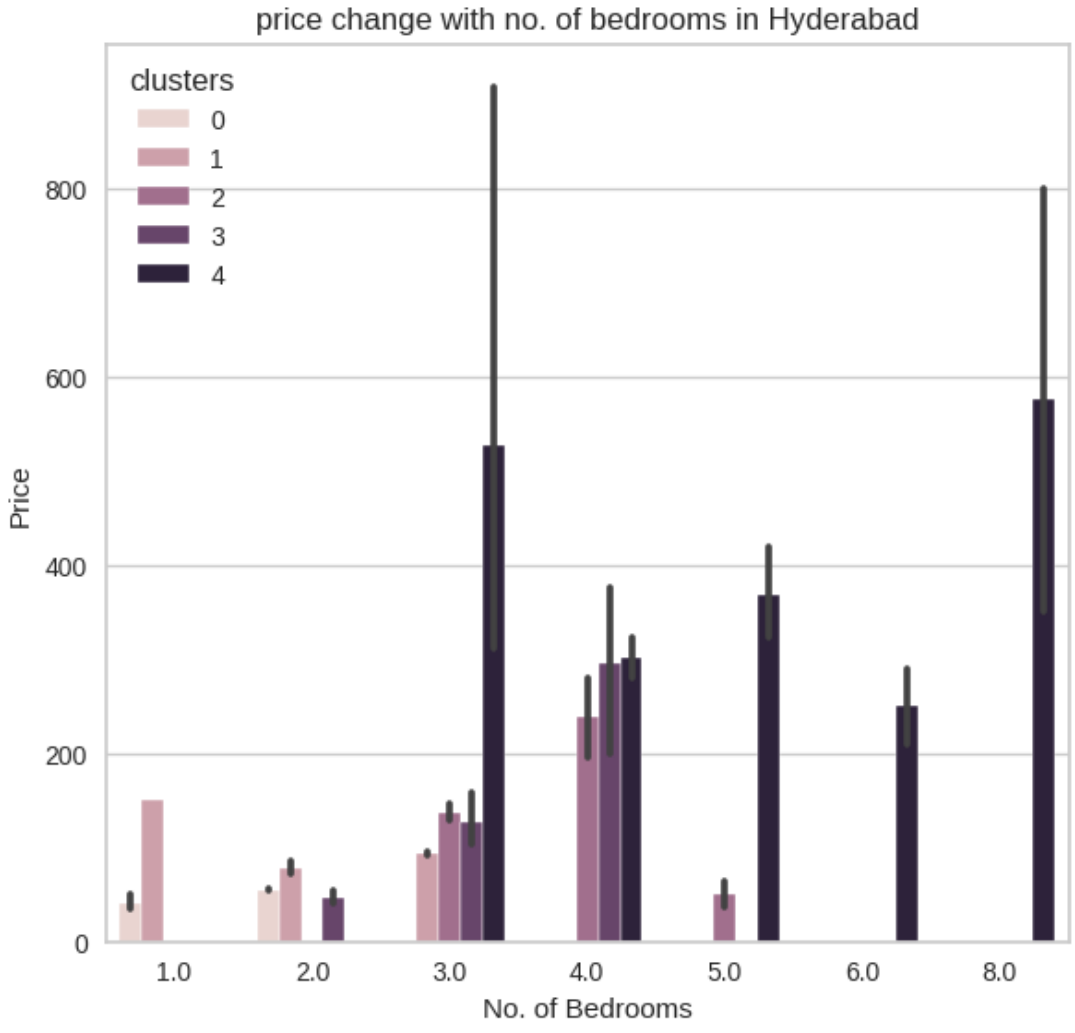
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='No. of Bedrooms', y='Price', hue=df['clusters'], data=tmpdf)
```

```
plt.title(f"price change with no. of bedrooms in {i}")
plt.show()
```

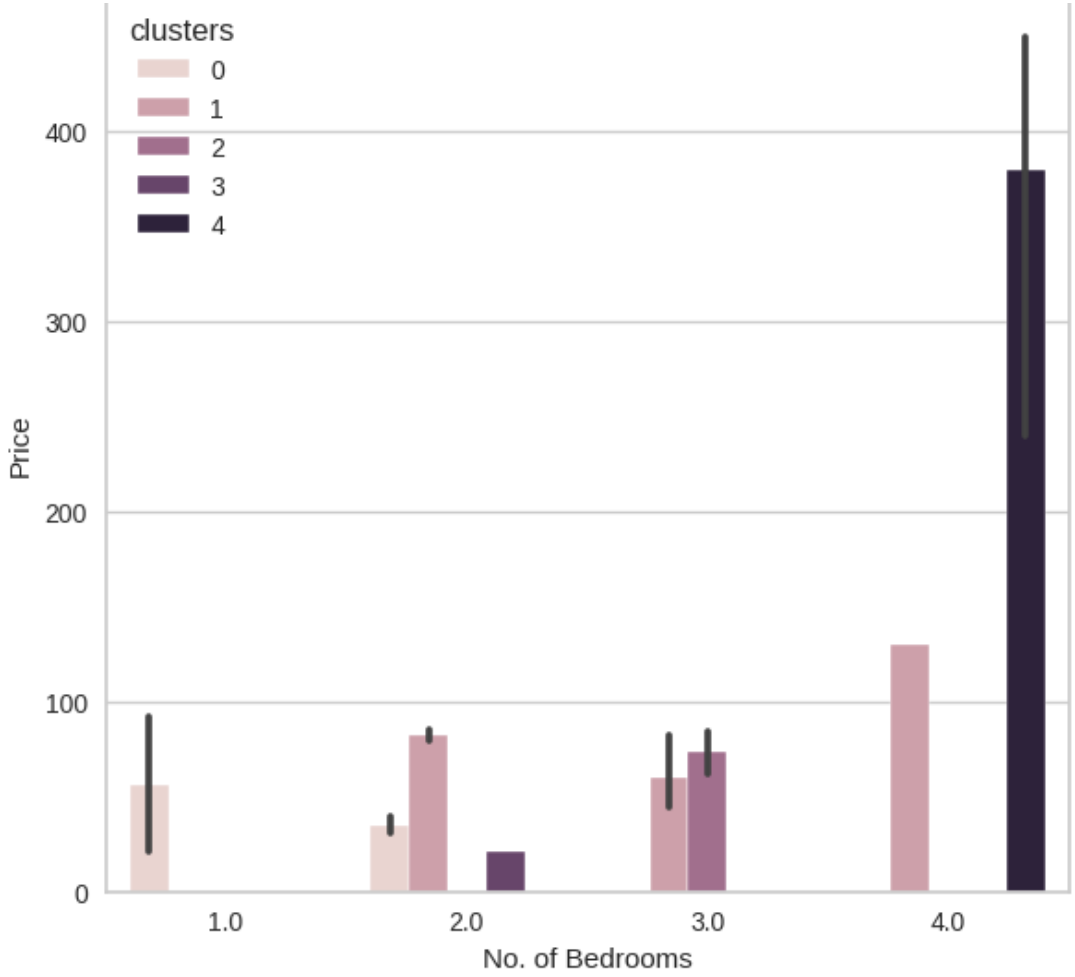


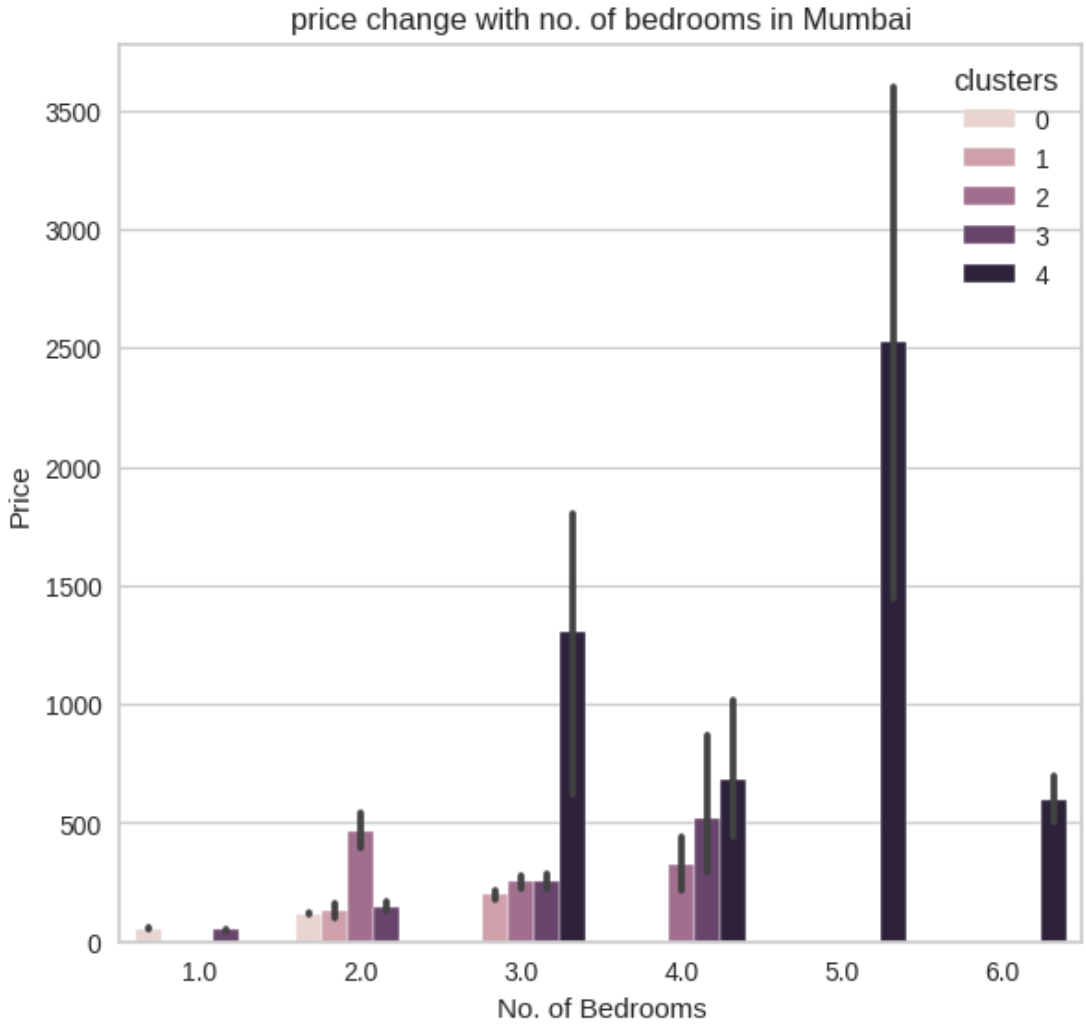






price change with no. of bedrooms in Kolkata

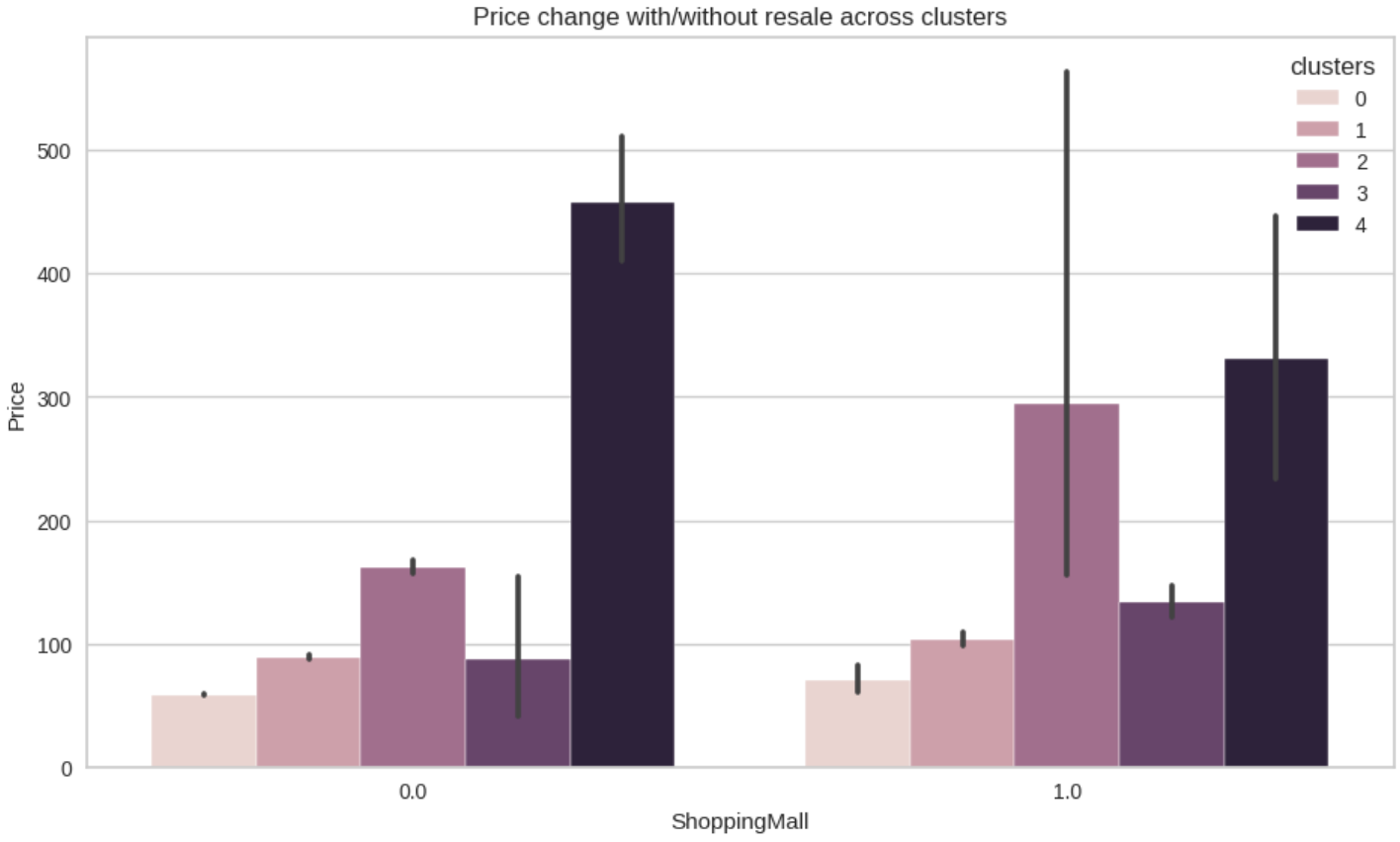




Shopping Mall

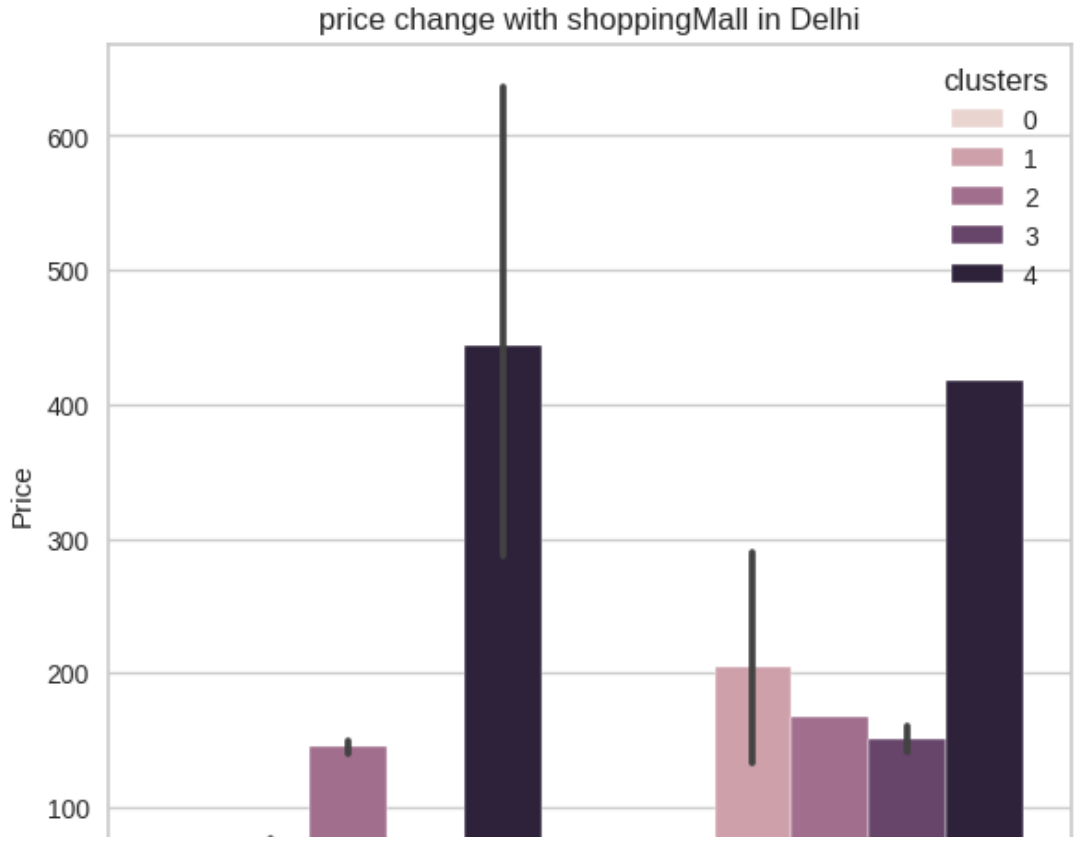
In [139]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['ShoppingMall'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with/without resale across clusters')
plt.show()
```

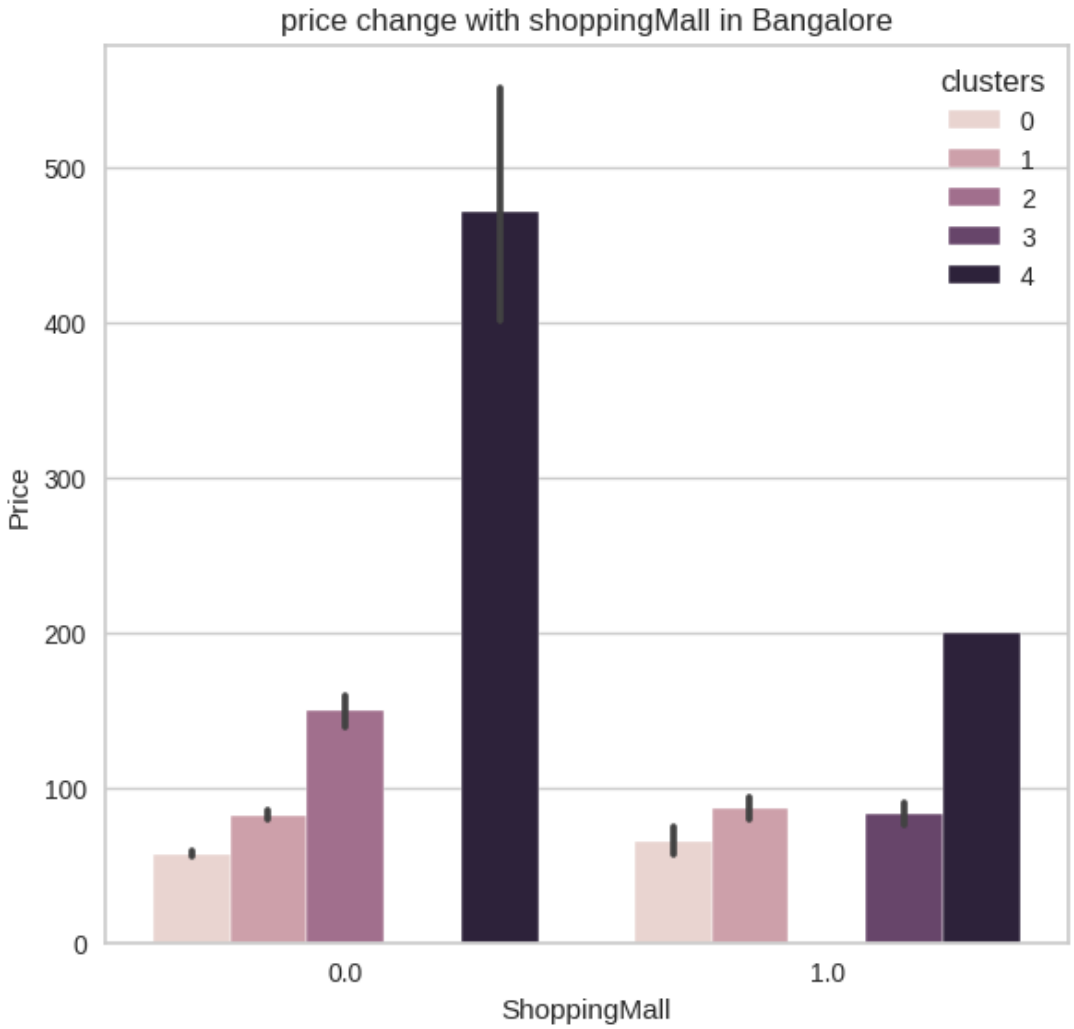


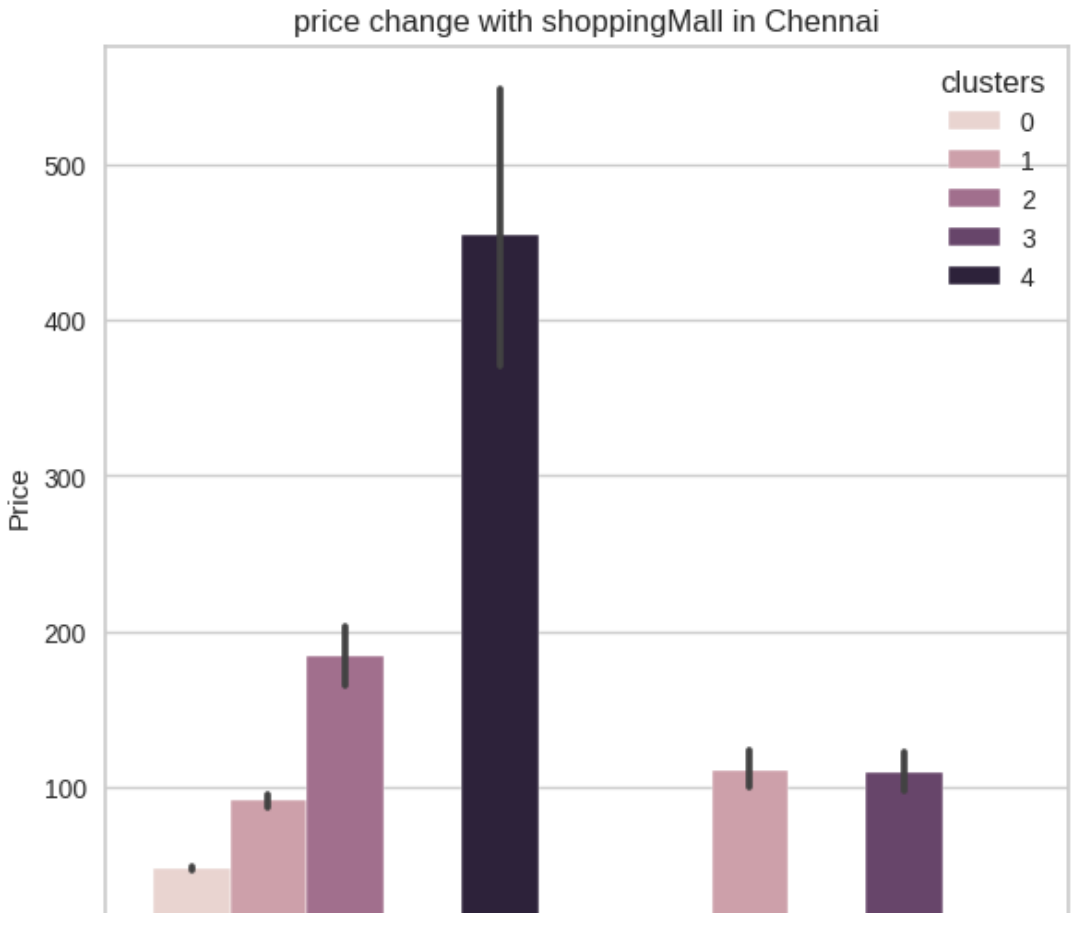
In [140]:

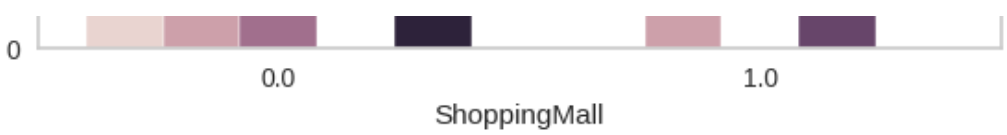
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='ShoppingMall', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with shoppingMall in {i}")
  plt.show()
```

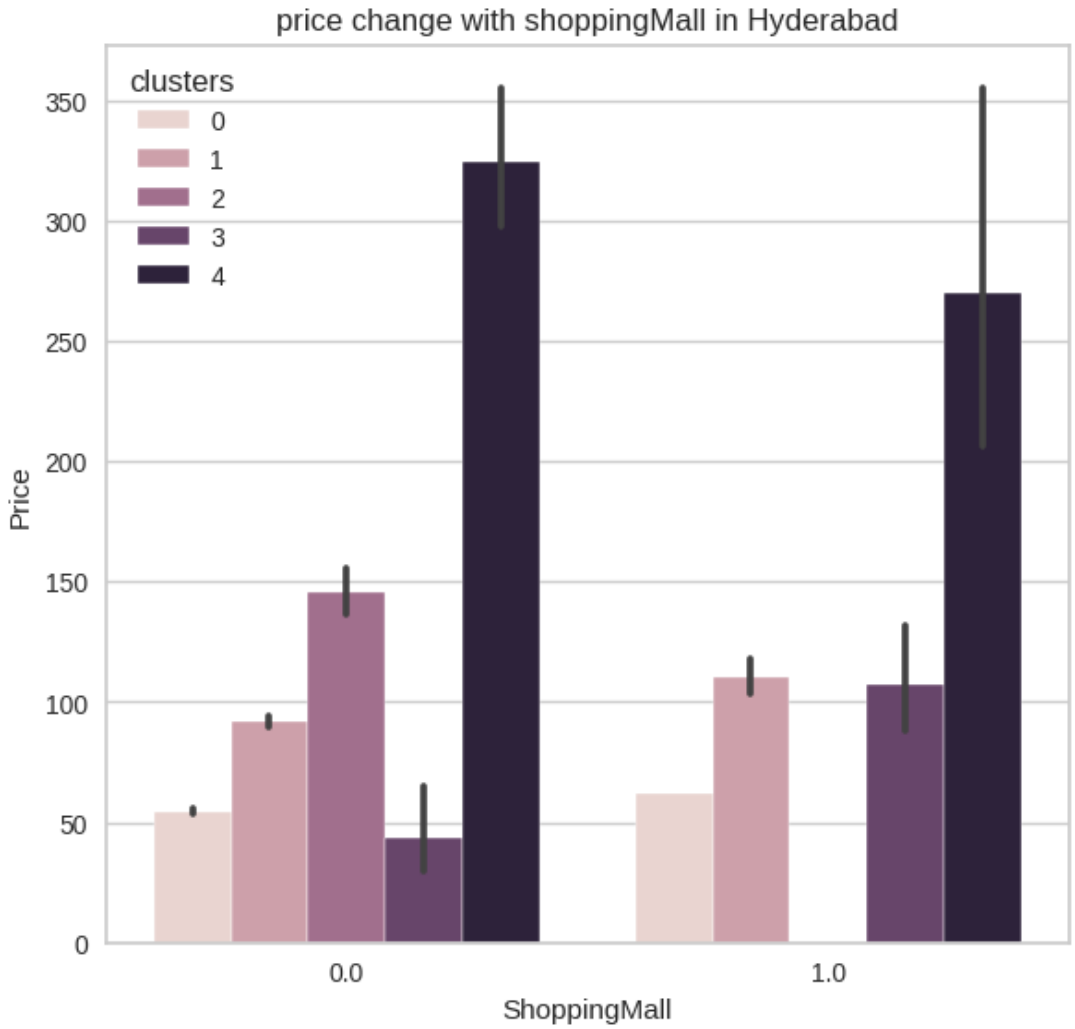


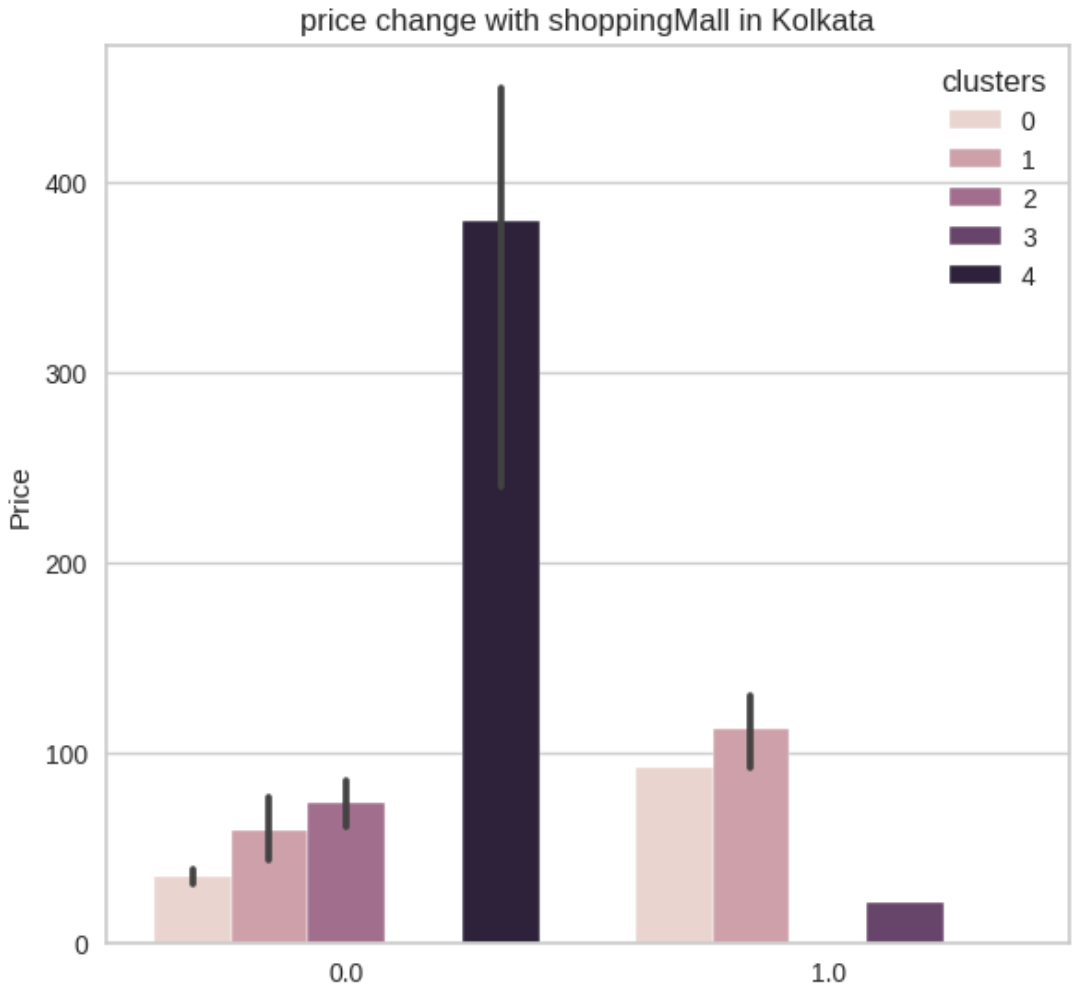

0
0.0
1.0
ShoppingMall

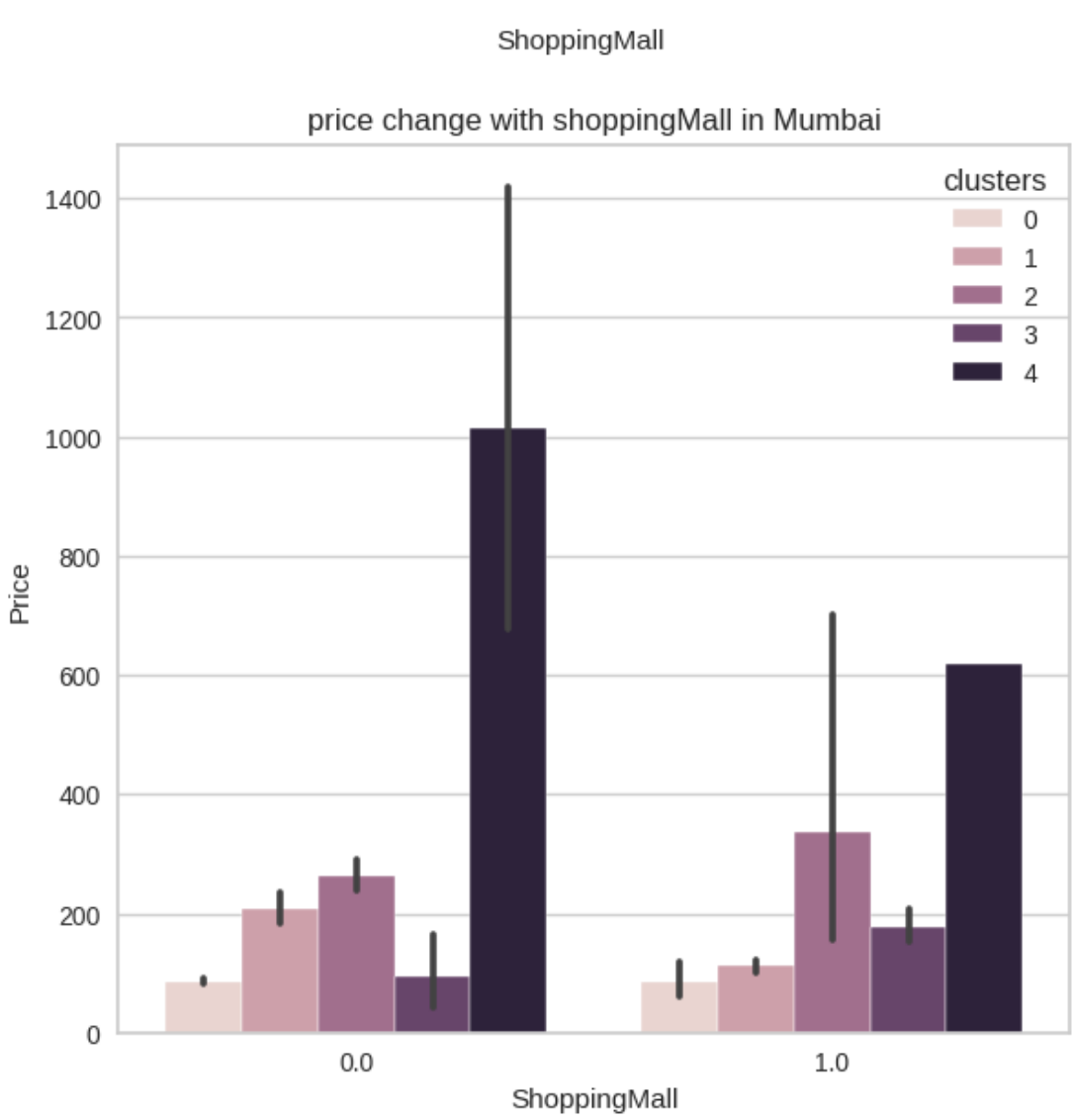

price change with shoppingMall in Bangalore
clusters
0
1
2
3
4
500
400
300
200
100
0
Price
0.0
1.0
ShoppingMall


price change with shoppingMall in Chennai
clusters
0
1
2
3
4
500
400
300
200
100
Price


0
0.0
1.0
ShoppingMall


price change with shoppingMall in Hyderabad
clusters
0
1
2
3
4
350
300
250
200
150
100
50
0
Price
0.0
1.0
ShoppingMall

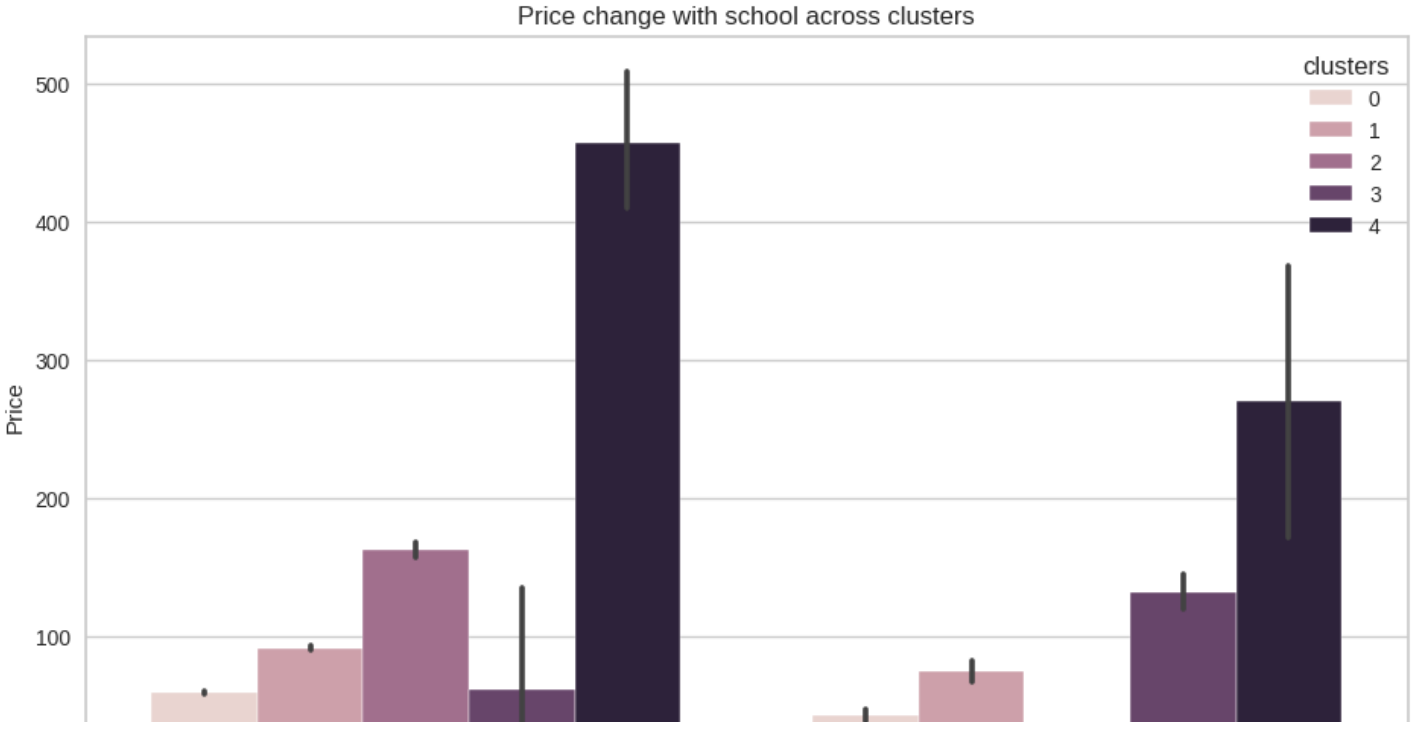

price change with shoppingMall in Kolkata
clusters
0
1
2
3
4
400
300
200
100
0
Price
0.0
1.0

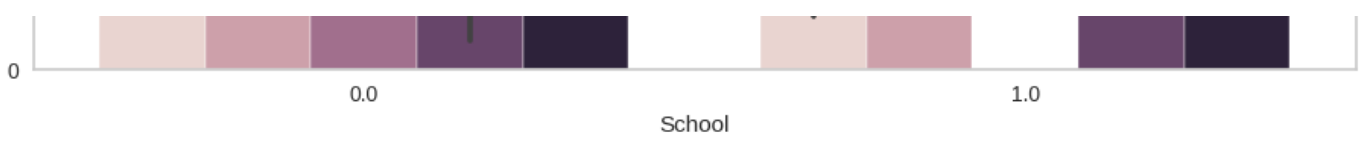


School

In [142]:

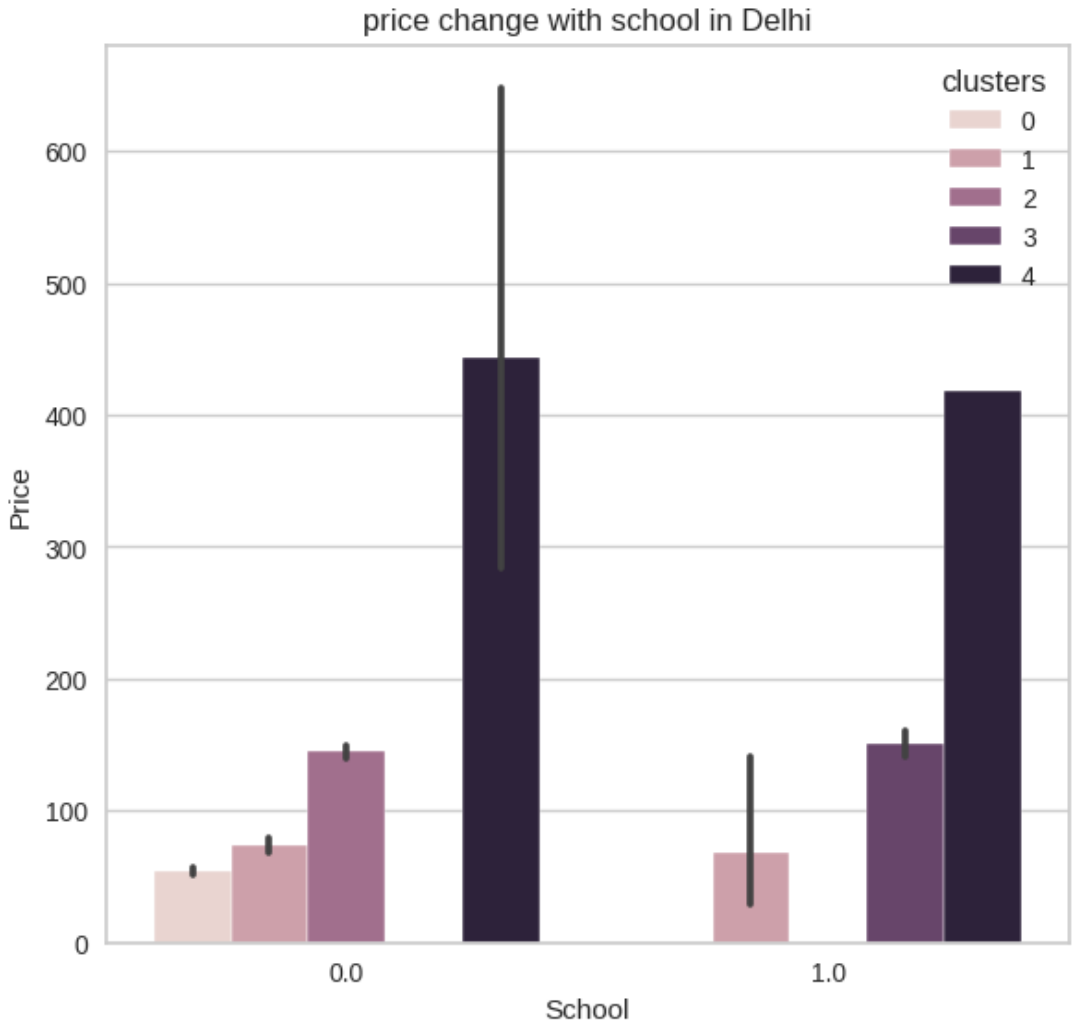
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['School'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with school across clusters')
plt.show()
```

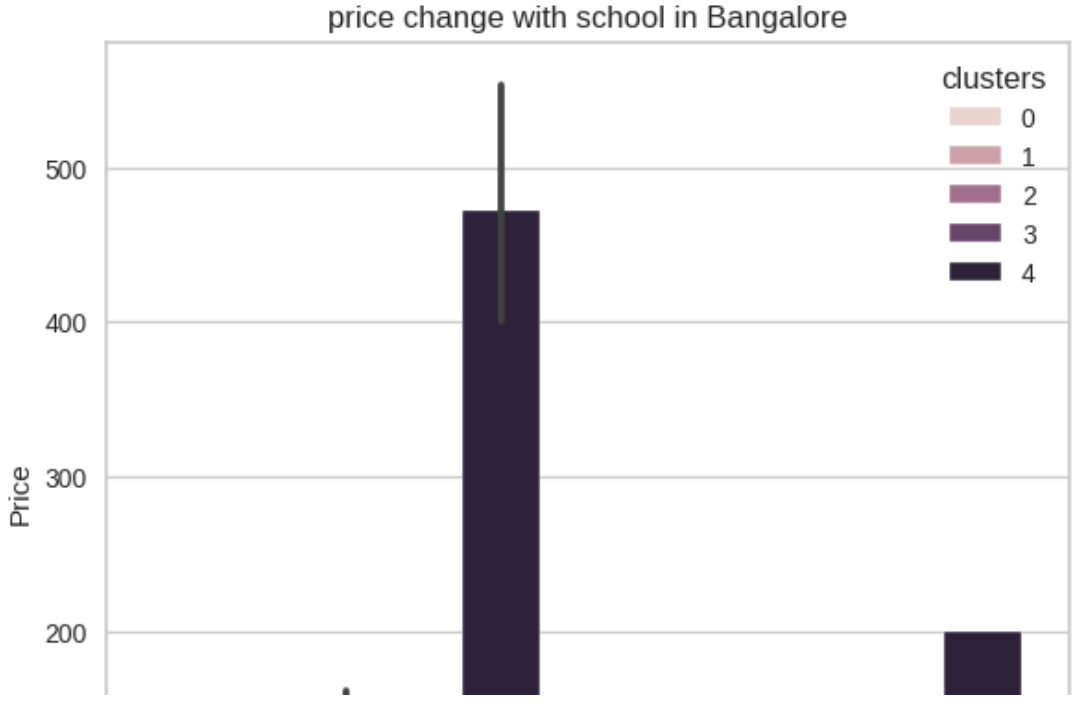


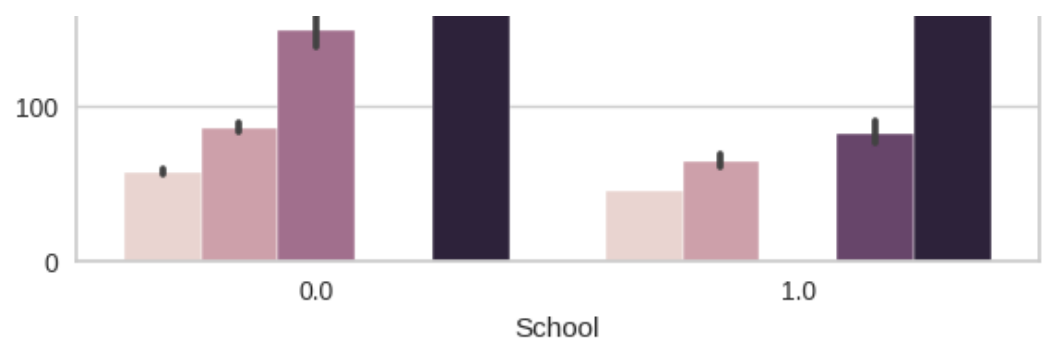


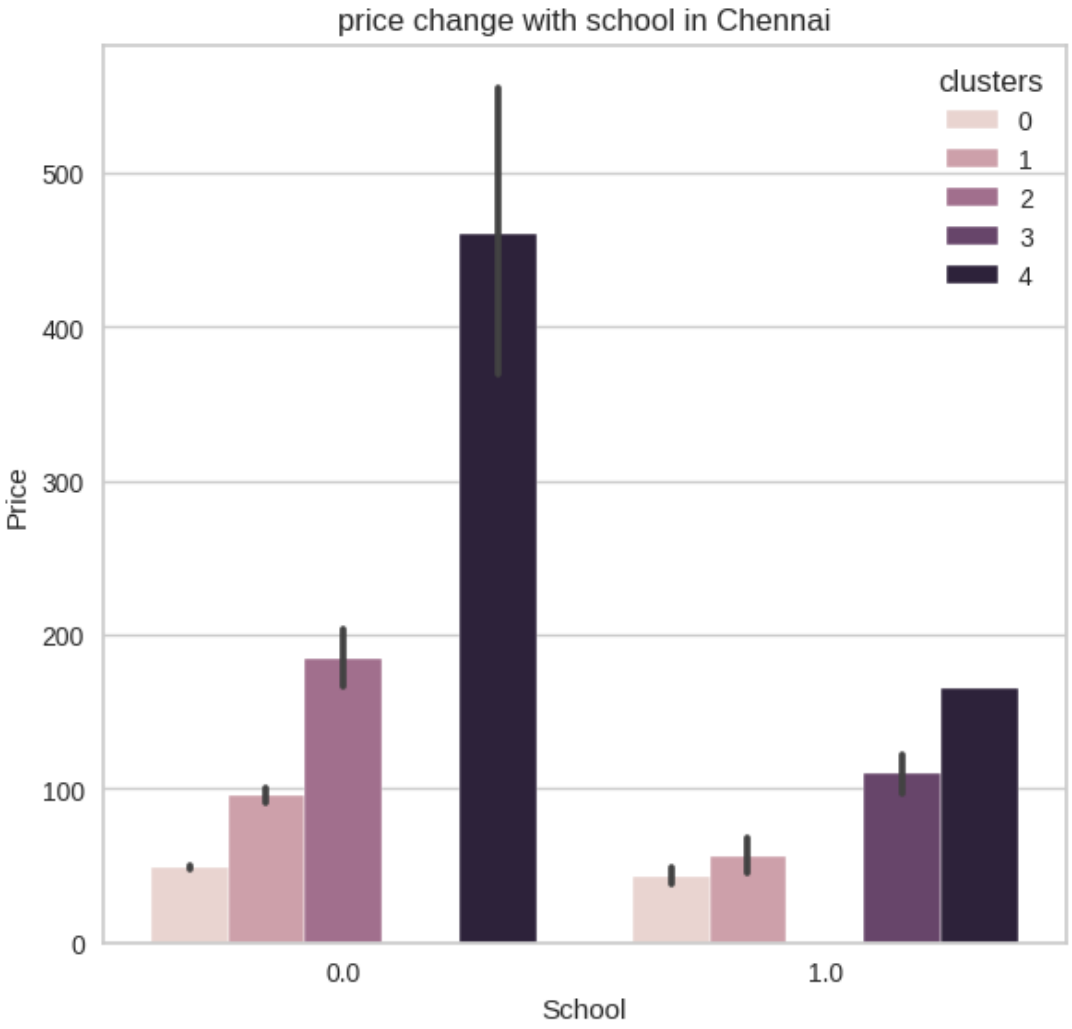
In [143]:

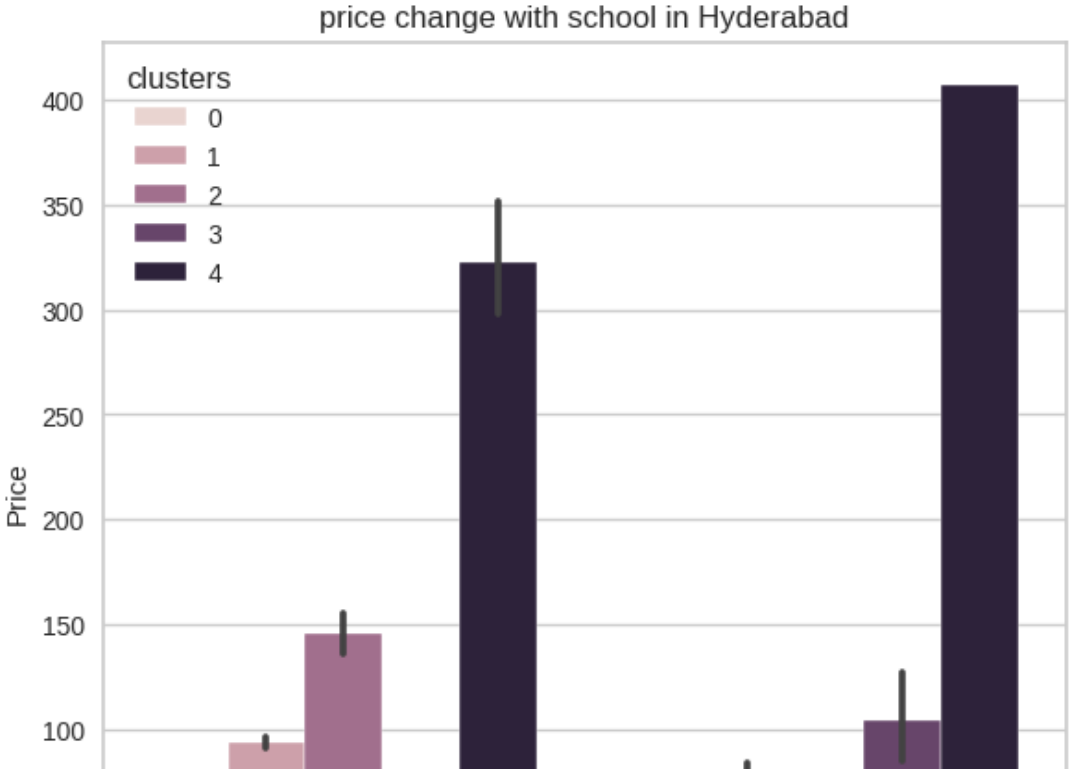
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='School', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with school in {i}")
  plt.show()
```

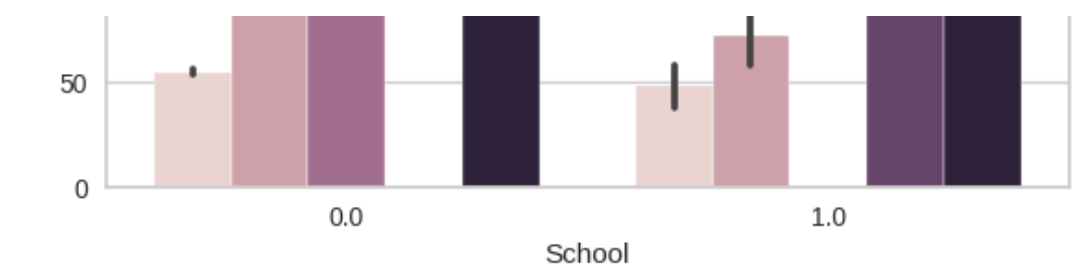


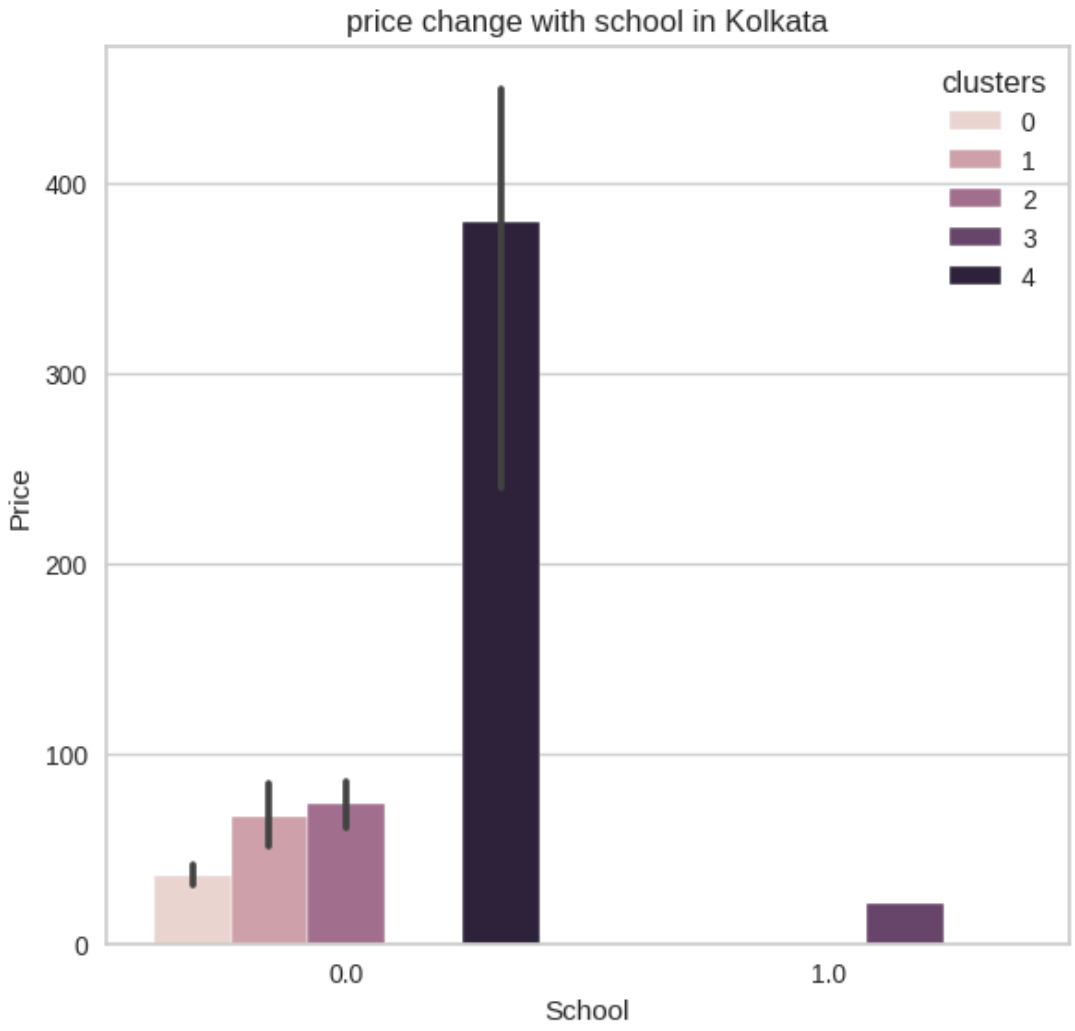


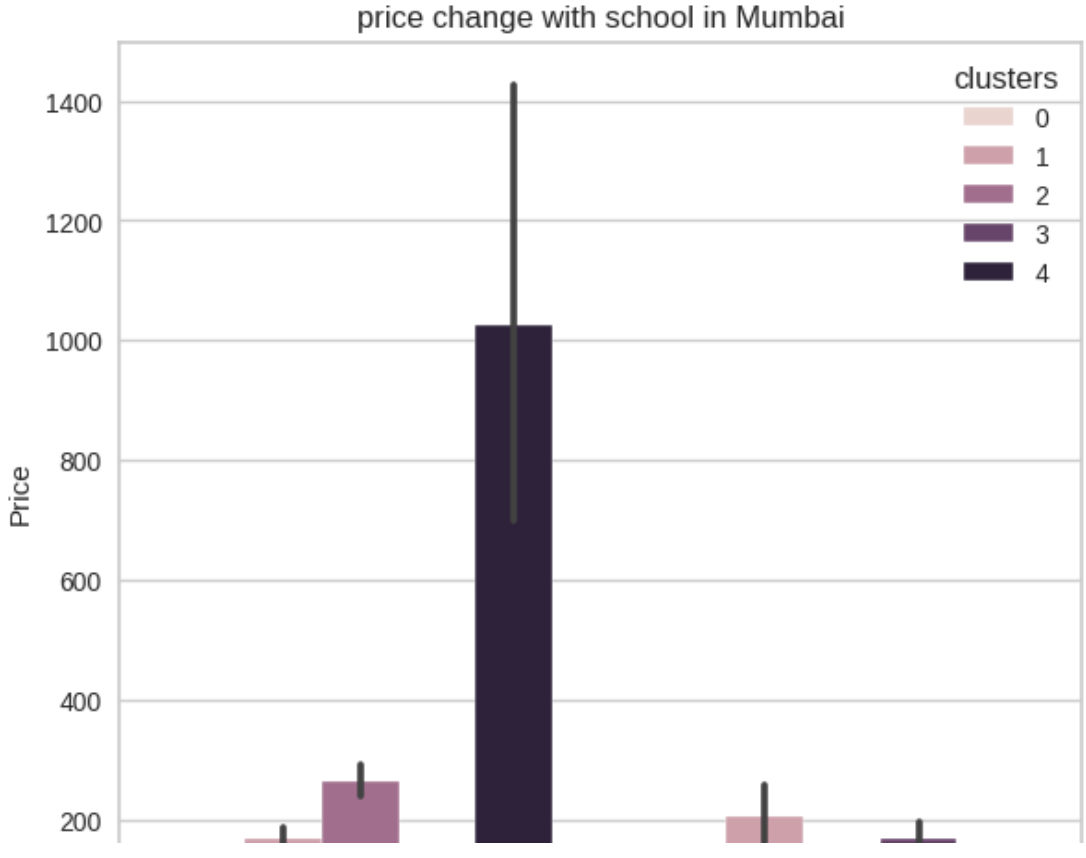

100
0
0.0
1.0
School

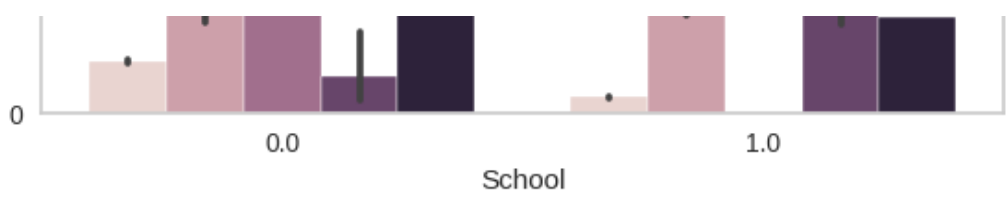

price change with school in Chennai
clusters
0
1
2
3
4
500
400
300
200
100
0
Price
0.0
1.0
School


price change with school in Hyderabad
clusters
0
1
2
3
4
400
350
300
250
200
150
100
Price


50
0
0.0
1.0
School


price change with school in Kolkata
clusters
0
1
2
3
4
400
300
200
100
0
Price
0.0
1.0
School

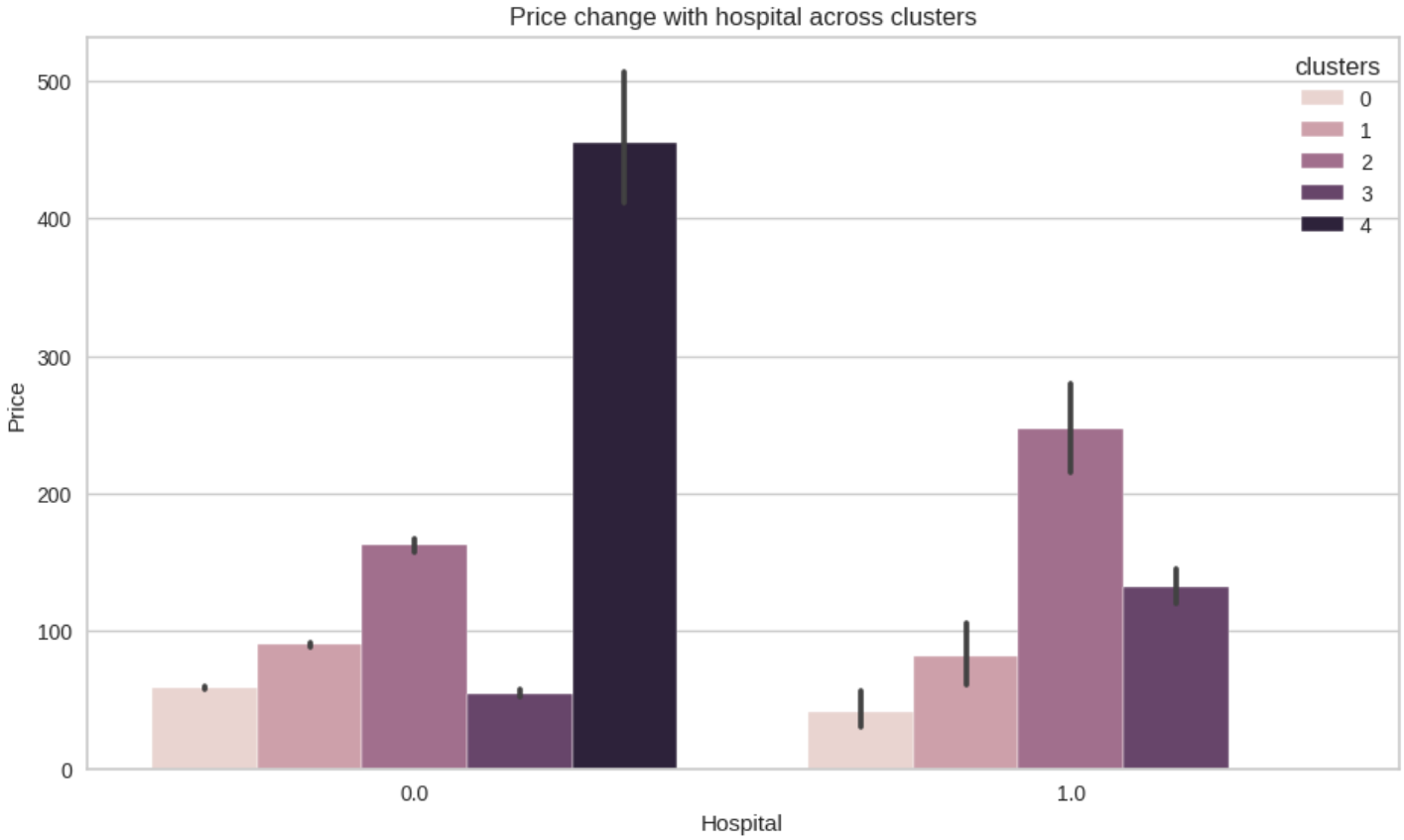

price change with school in Mumbai
clusters
0
1
2
3
4
1400
1200
1000
800
600
400
200
Price



Hospital

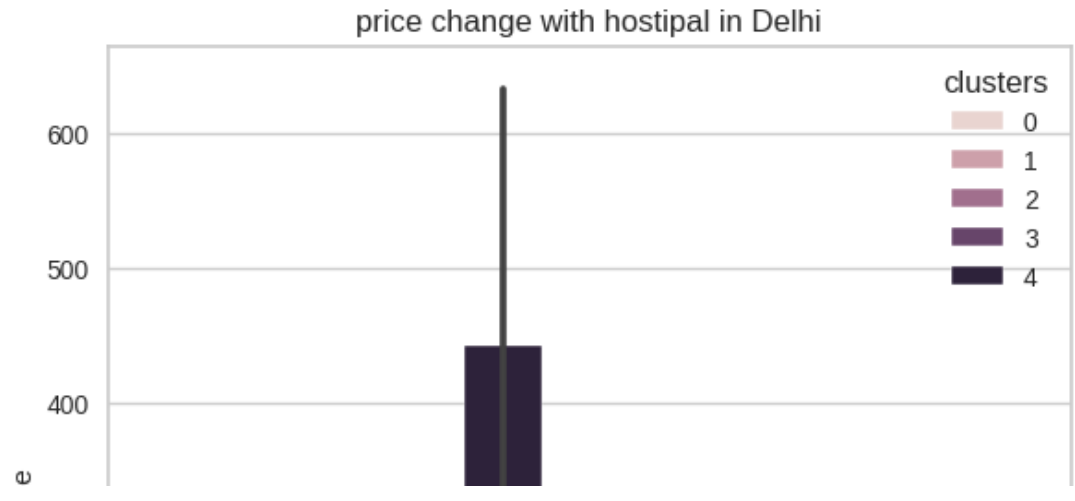
In [145]:

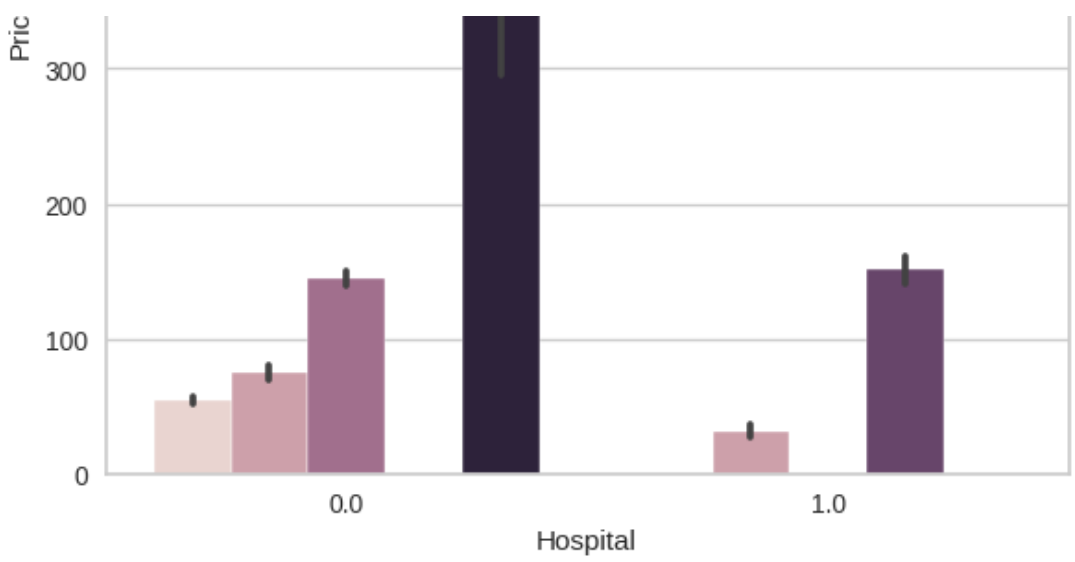
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['Hospital'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with hospital across clusters')
plt.show()
```

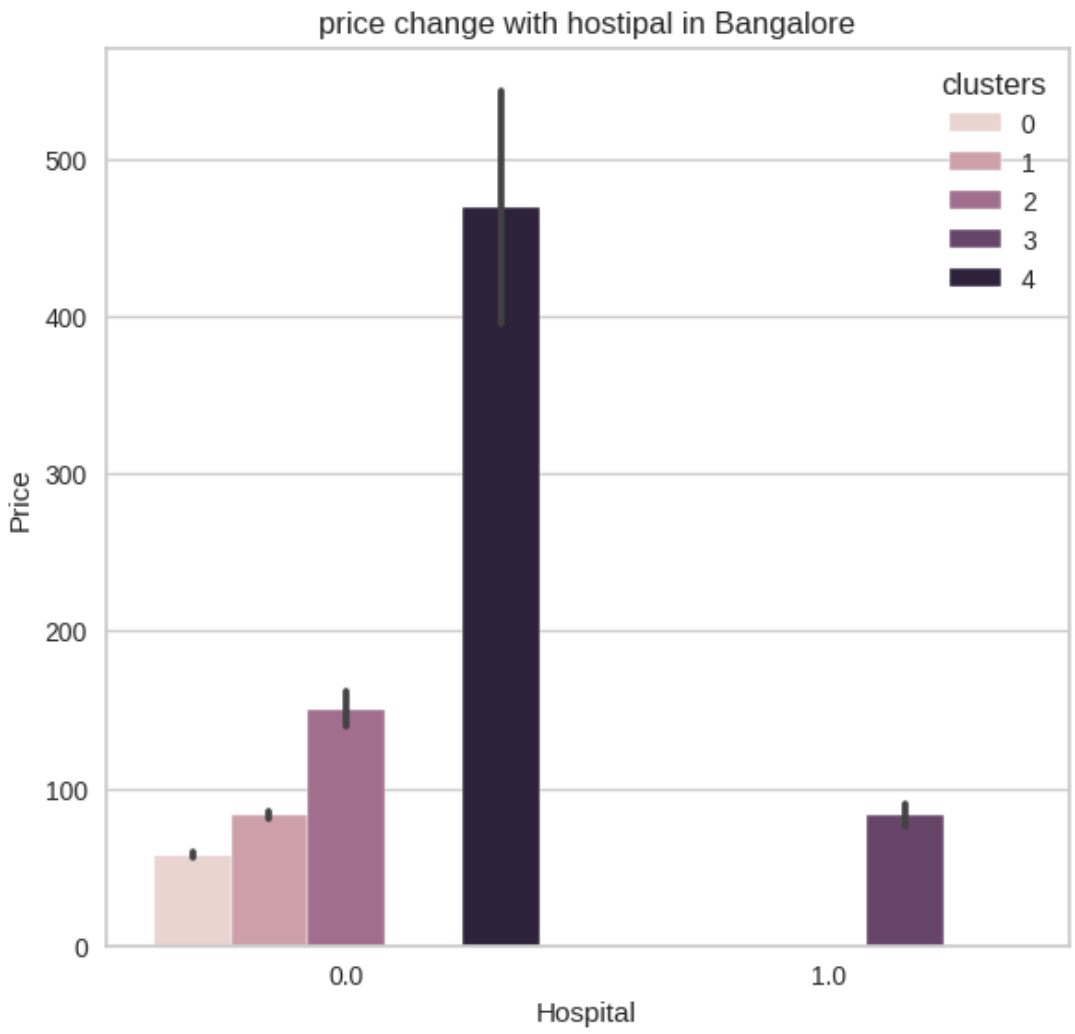


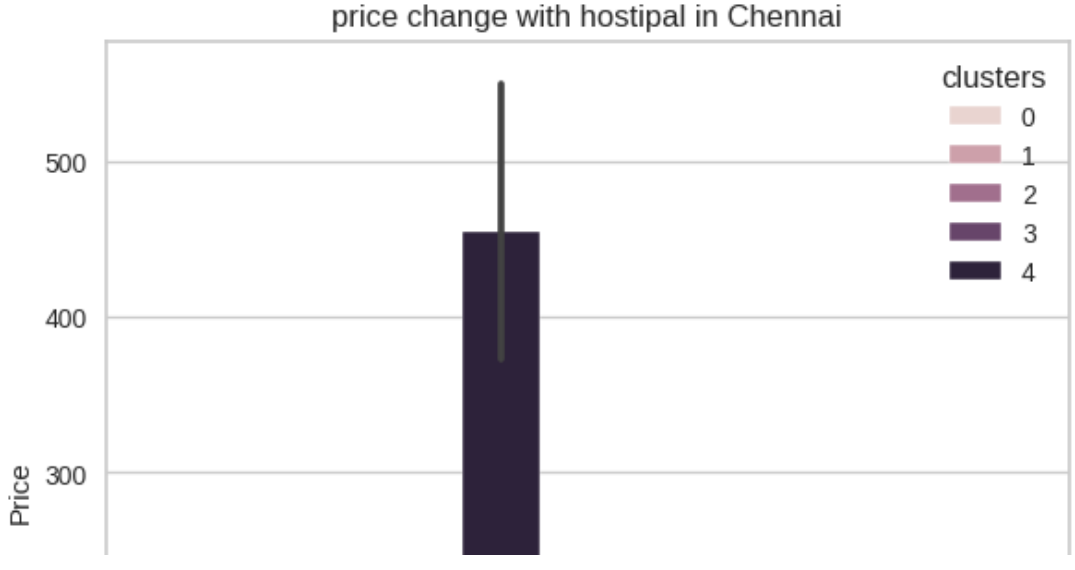
In [146]:

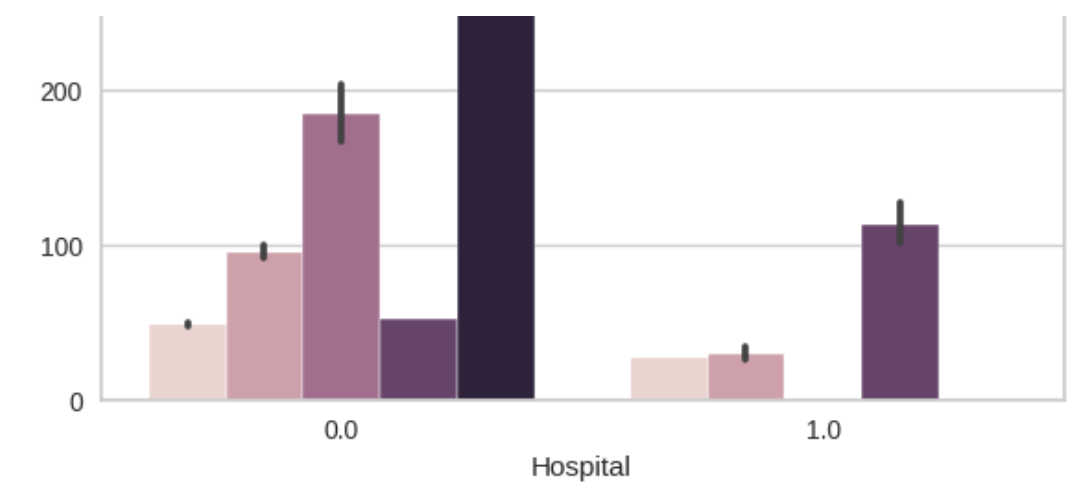
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='Hospital', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with hostipal in {i}")
  plt.show()
```



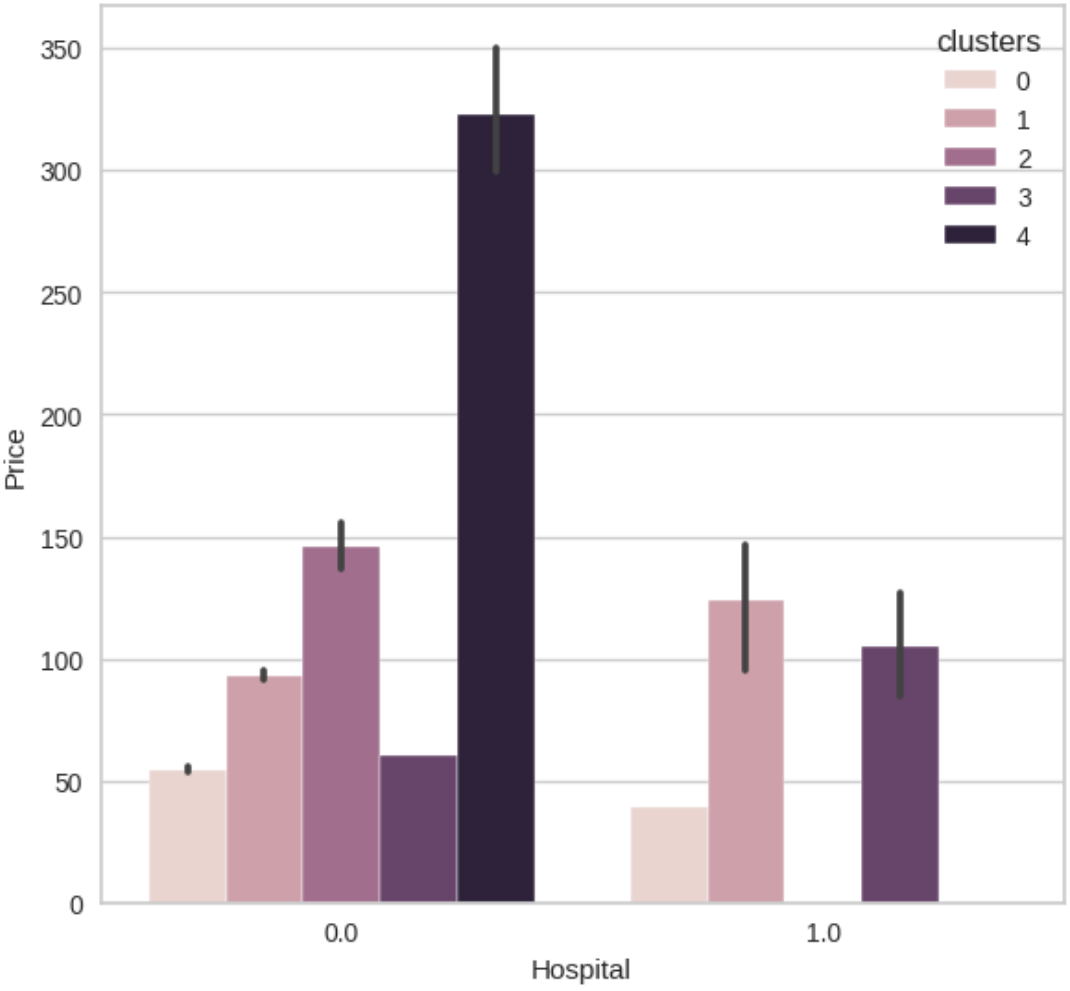

Pric
300
200
100
0
0.0
1.0
Hospital


price change with hostipal in Bangalore
clusters
0
1
2
3
4
500
400
300
200
100
0
Price
0.0
1.0
Hospital

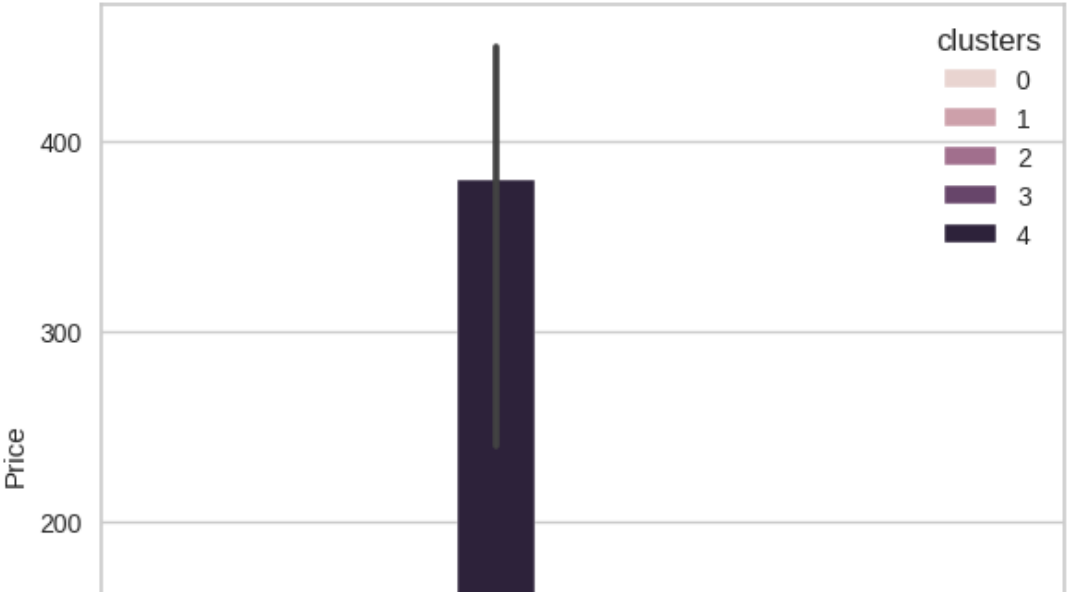

price change with hostipal in Chennai
clusters
0
1
2
3
4
500
400
300
Price

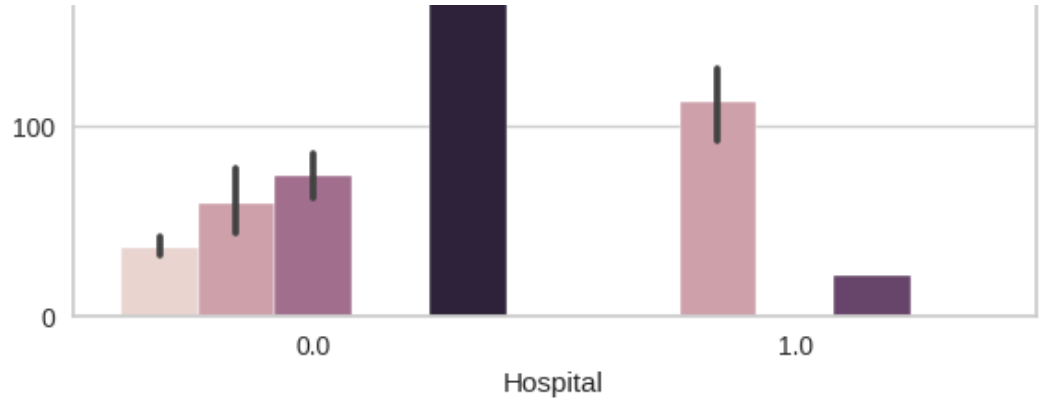


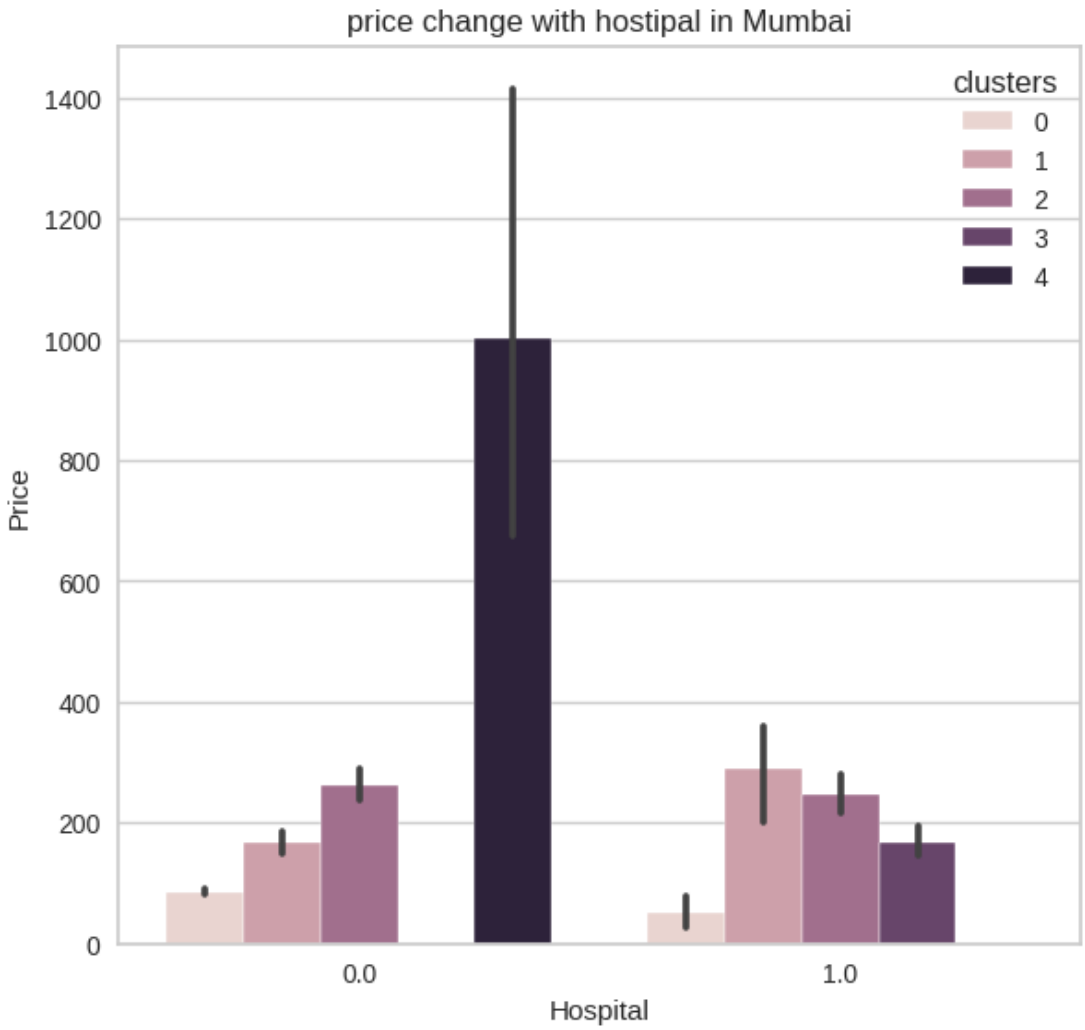
price change with hostipal in Hyderabad



price change with hostipal in Kolkata



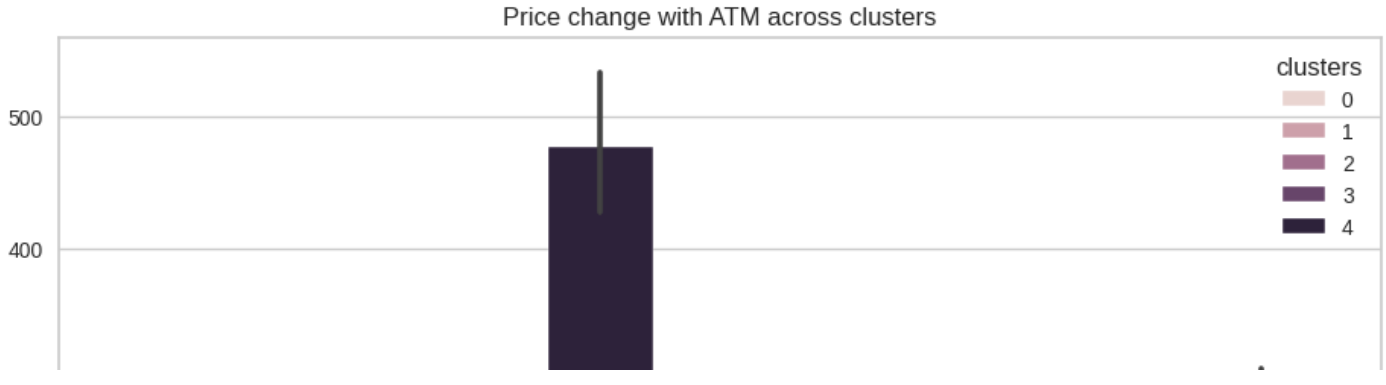




ATM

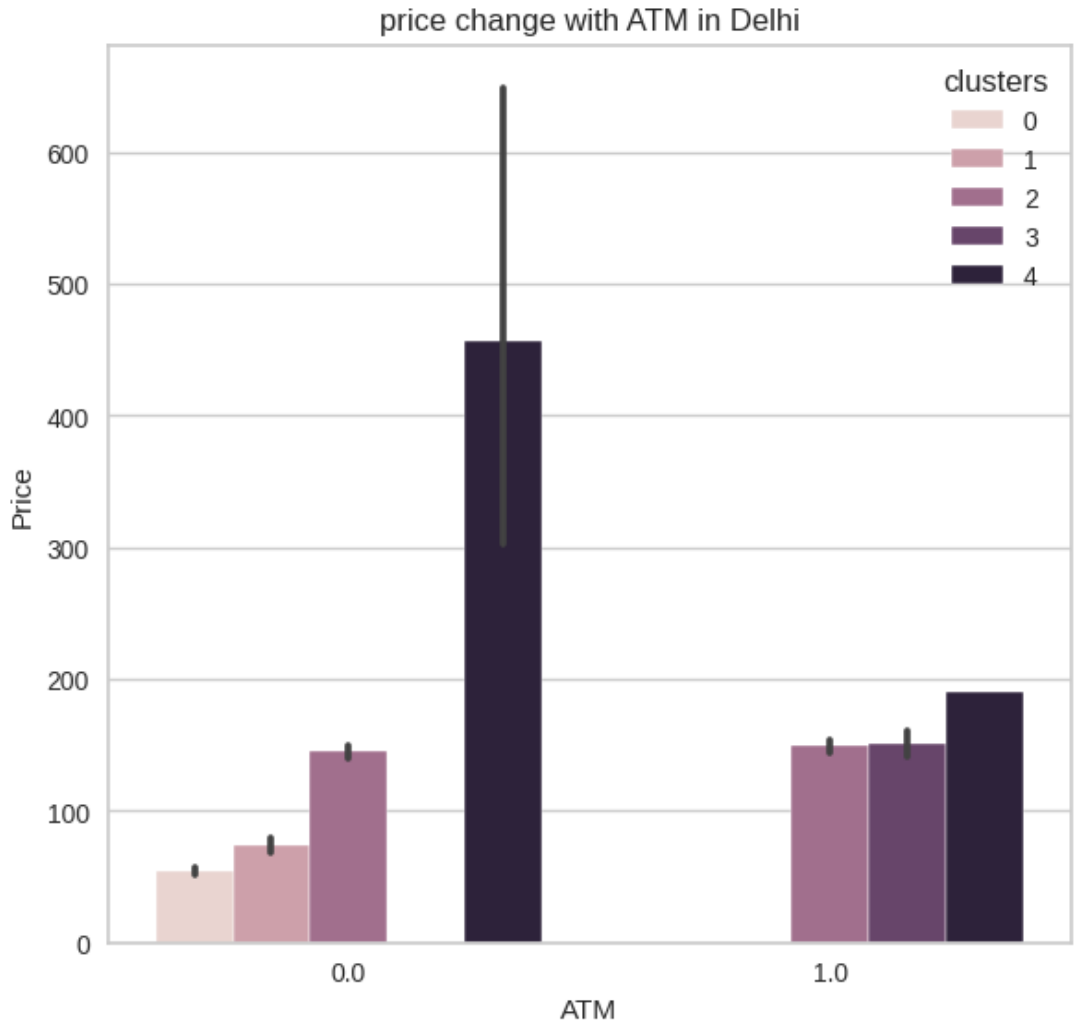
In [148]:

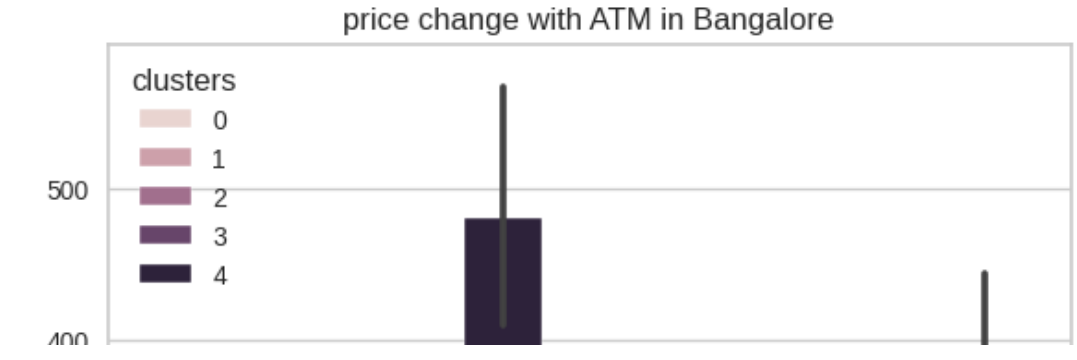
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['ATM'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with ATM across clusters')
plt.show()
```

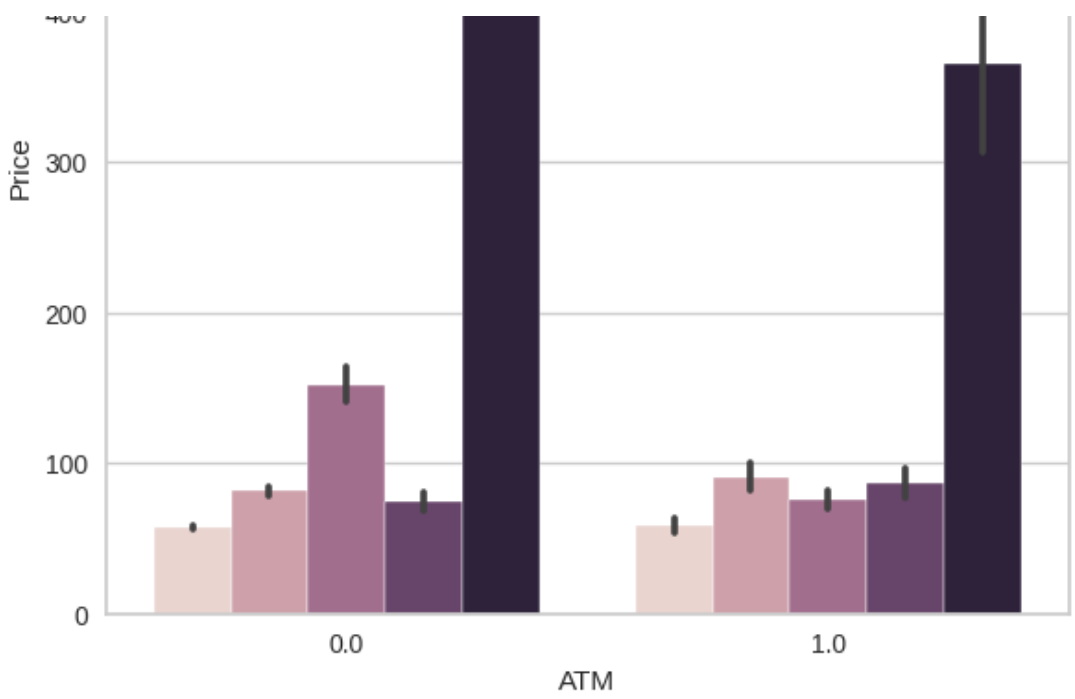


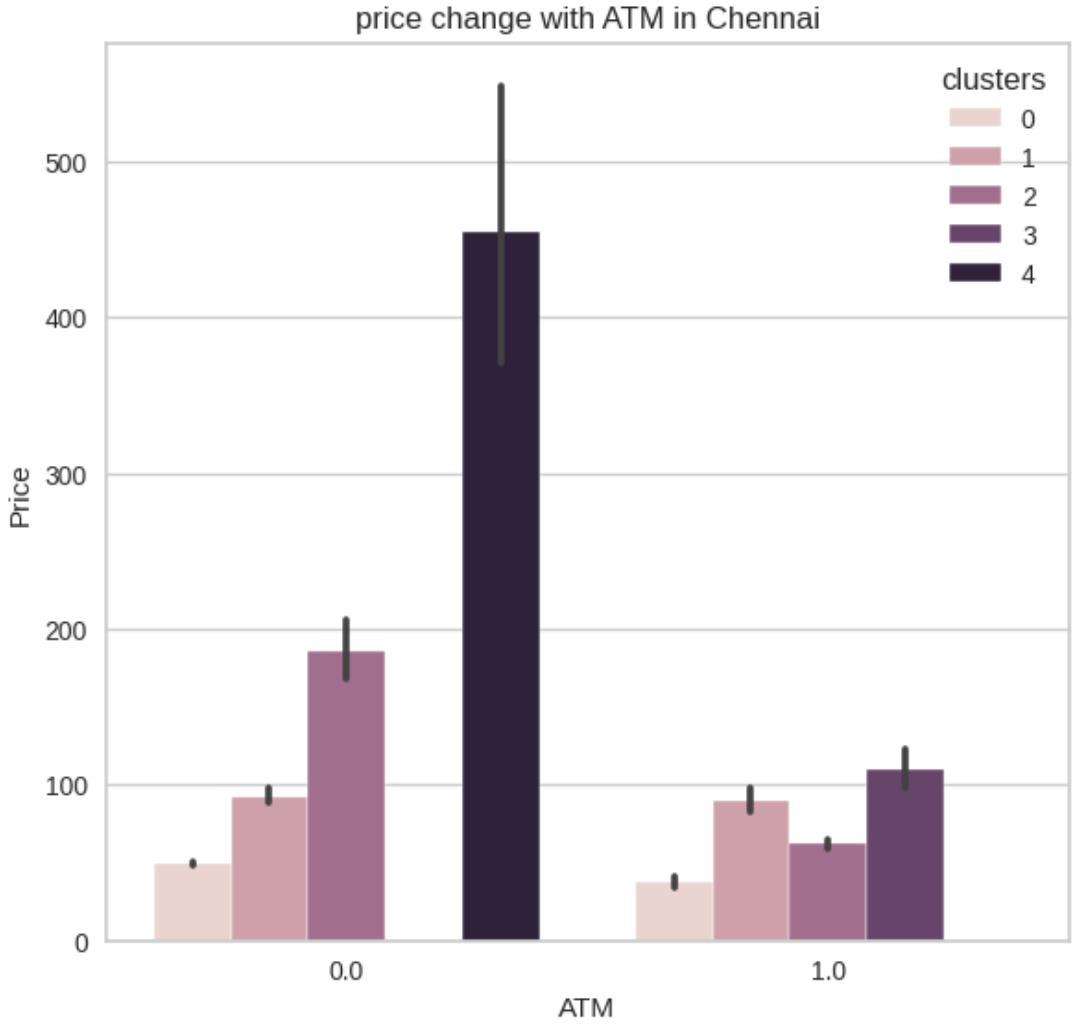
In [149]:

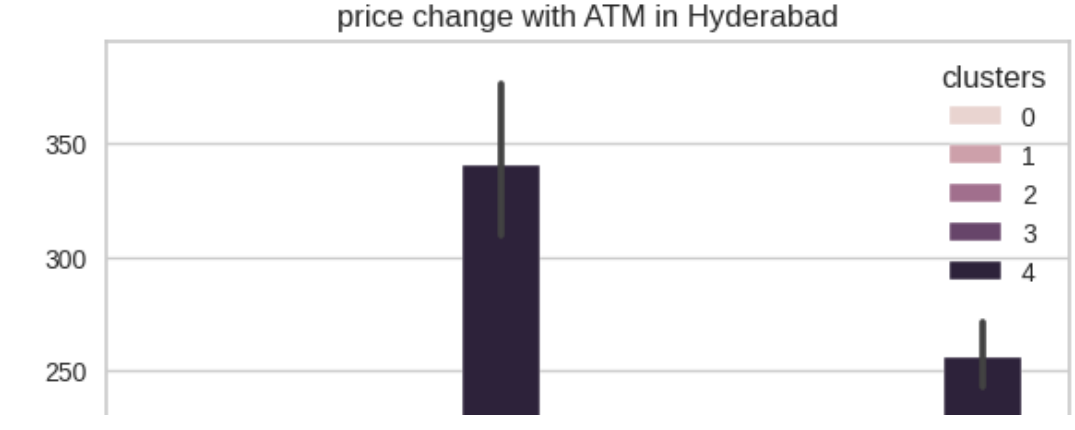
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='ATM', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with ATM in {i}")
  plt.show()
```

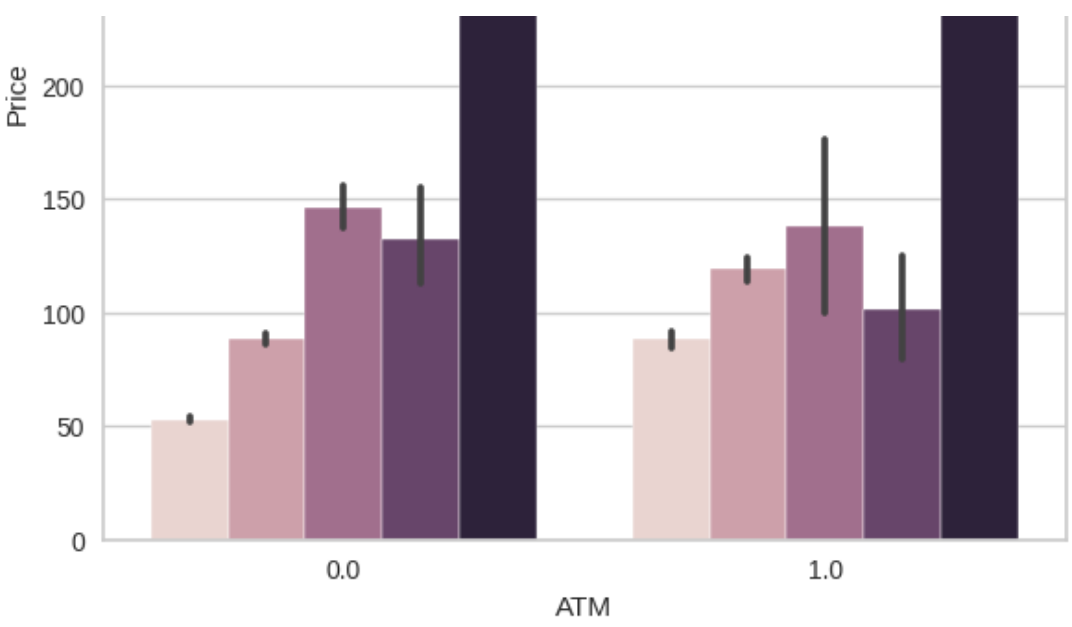


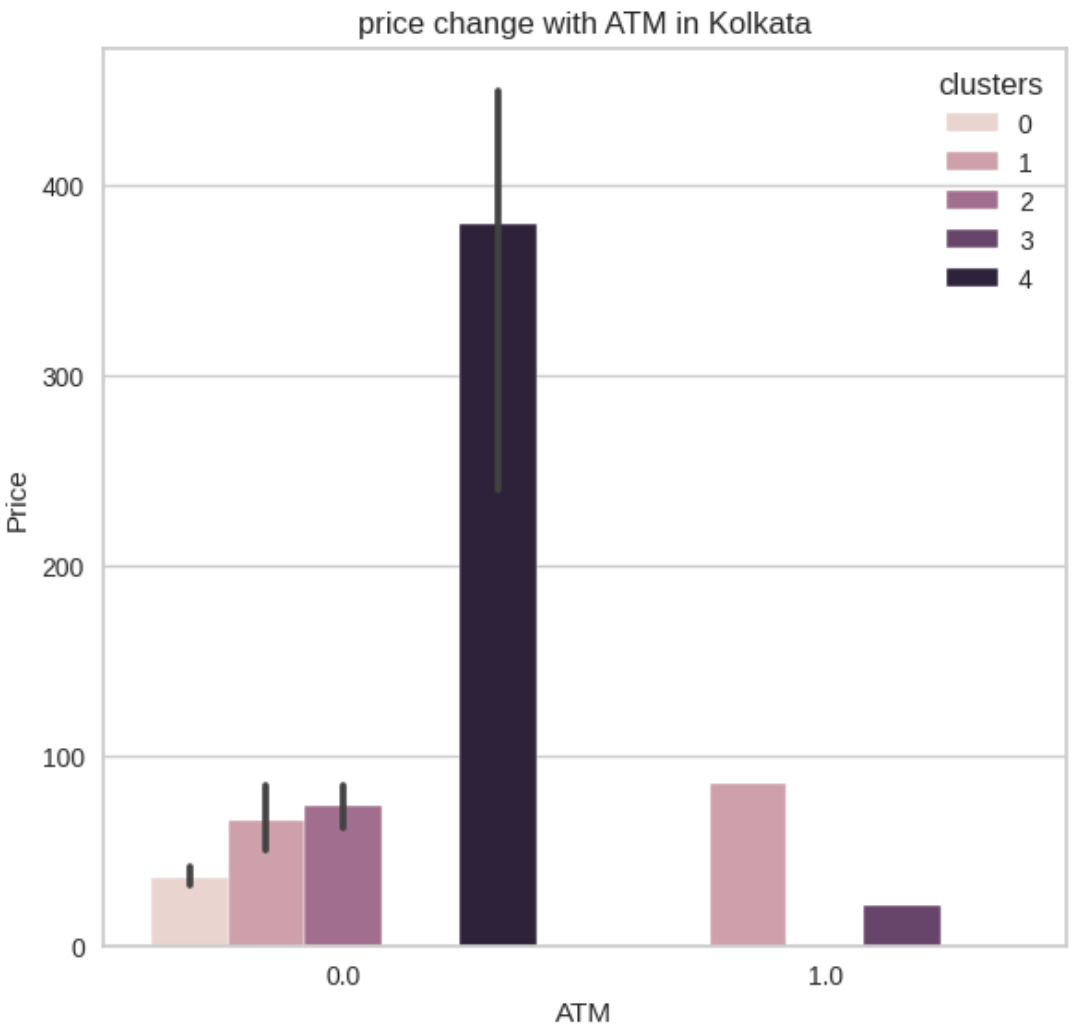


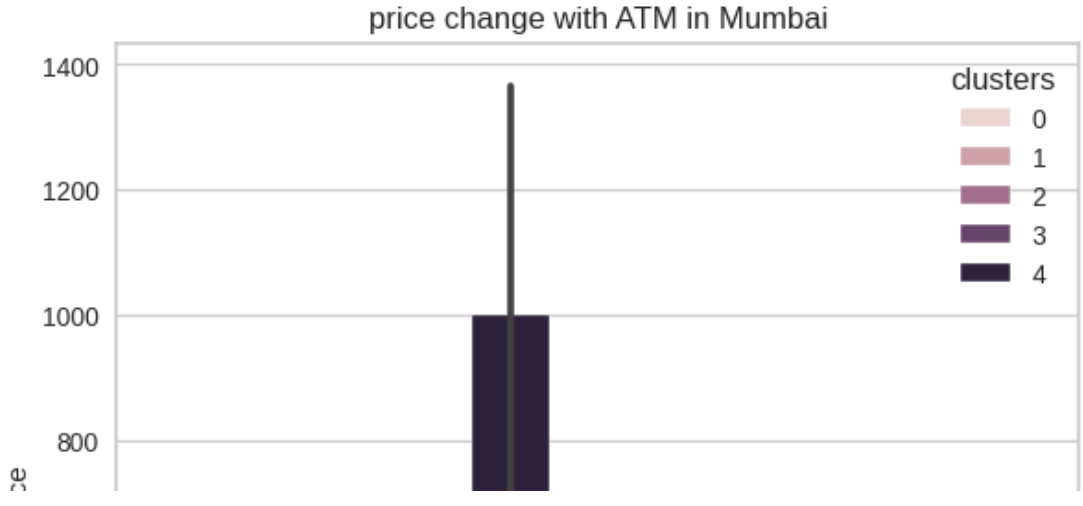

Price
300
200
100
0
0.0
1.0
ATM

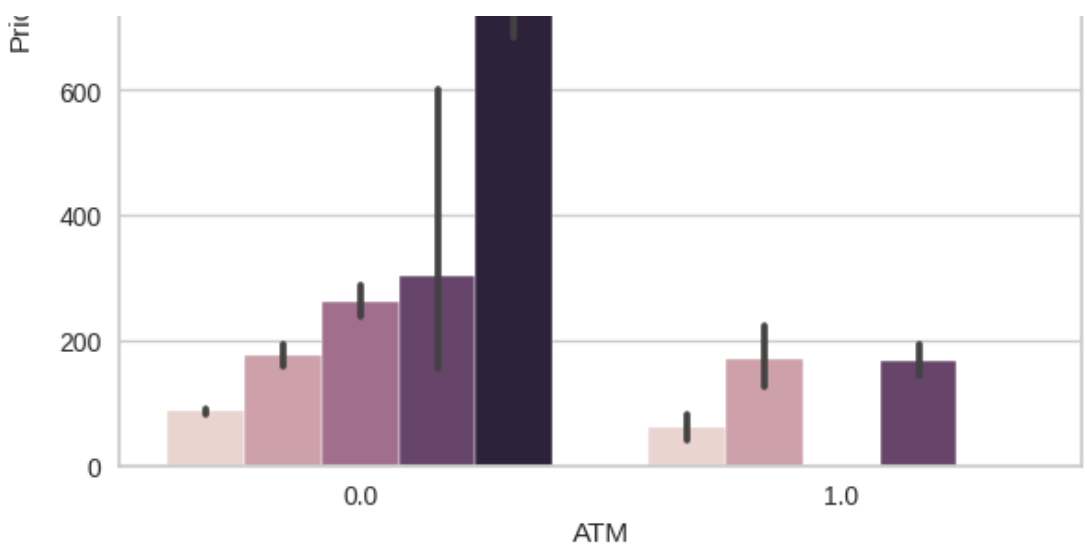

price change with ATM in Chennai
clusters
0
1
2
3
4
Price
500
400
300
200
100
0
0.0
1.0
ATM


price change with ATM in Hyderabad
clusters
0
1
2
3
4
350
300
250


Price
200
150
100
50
0
0.0
1.0
ATM


price change with ATM in Kolkata
clusters
0
1
2
3
4
Price
400
300
200
100
0
0.0
1.0
ATM

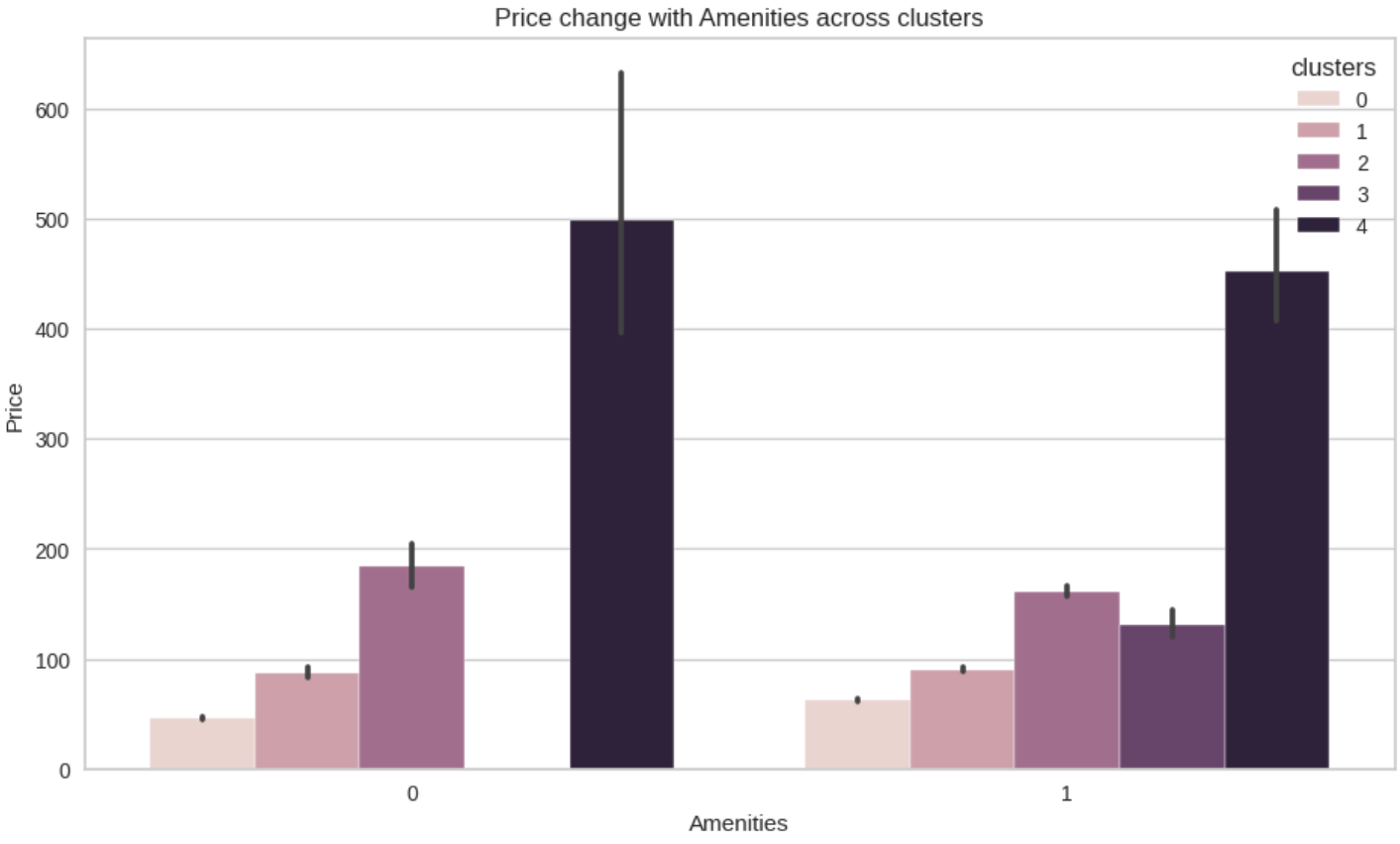

price change with ATM in Mumbai
clusters
0
1
2
3
4
1400
1200
1000
800



Amenities

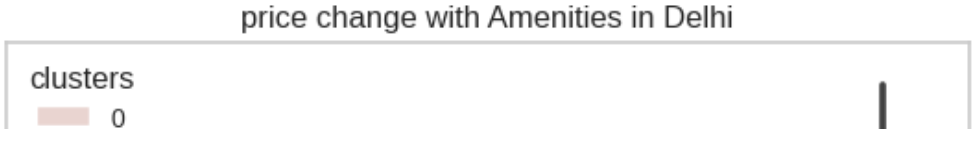
In [151]:

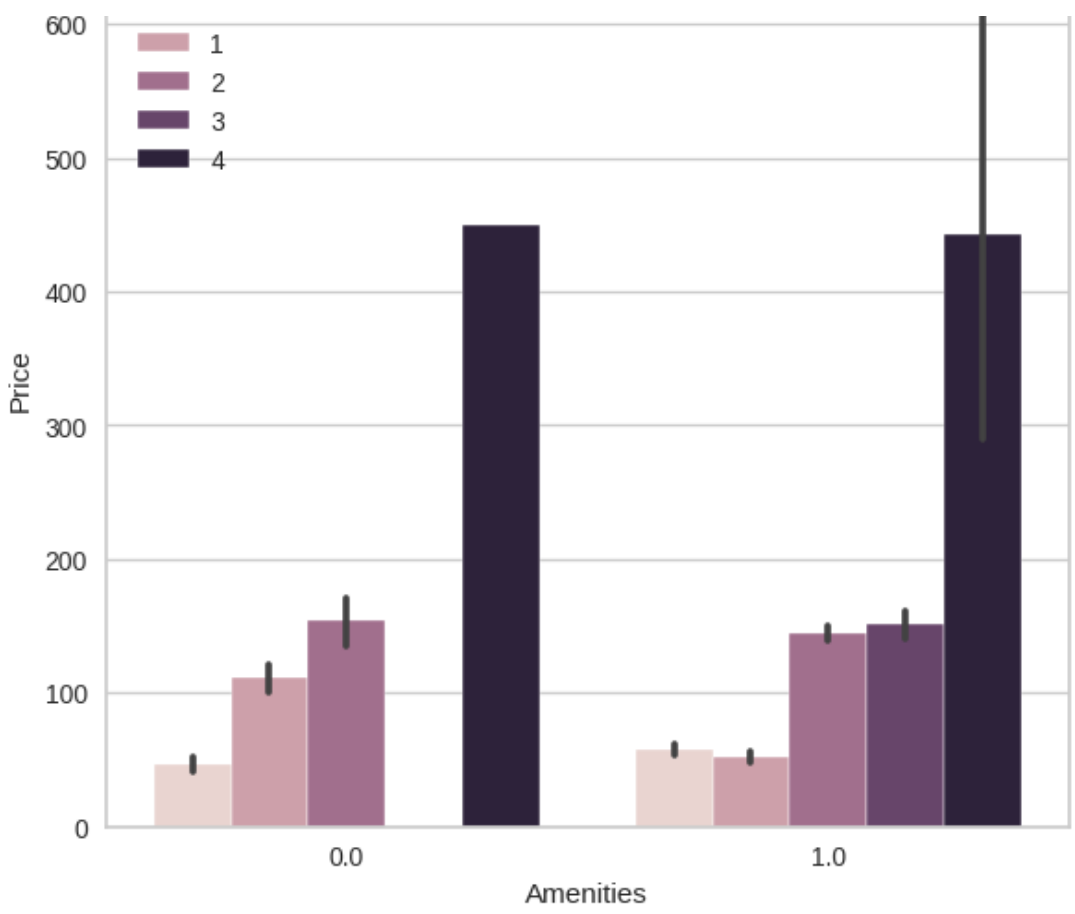
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['Amenities'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with Amenities across clusters')
plt.show()
```



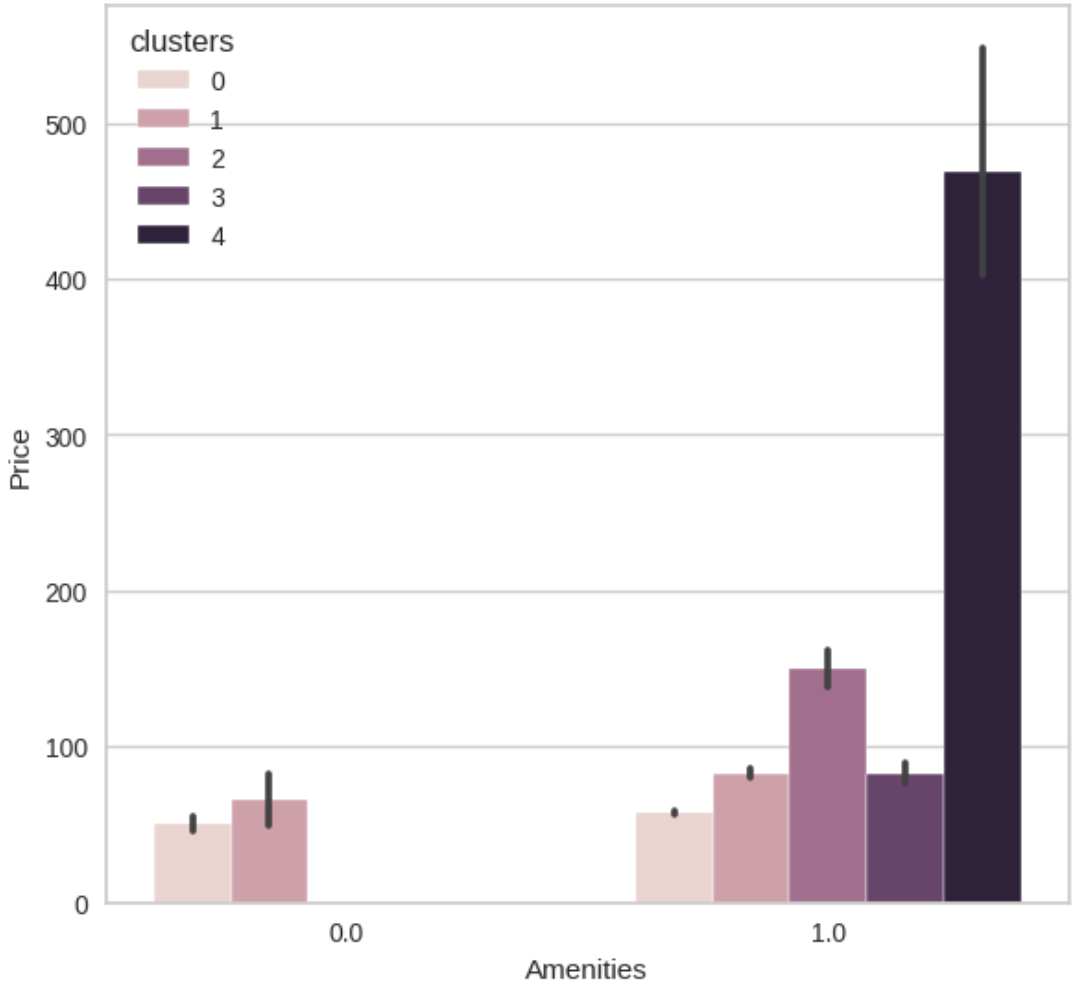
In [152]:

```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='Amenities', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with Amenities in {i}")
  plt.show()
```

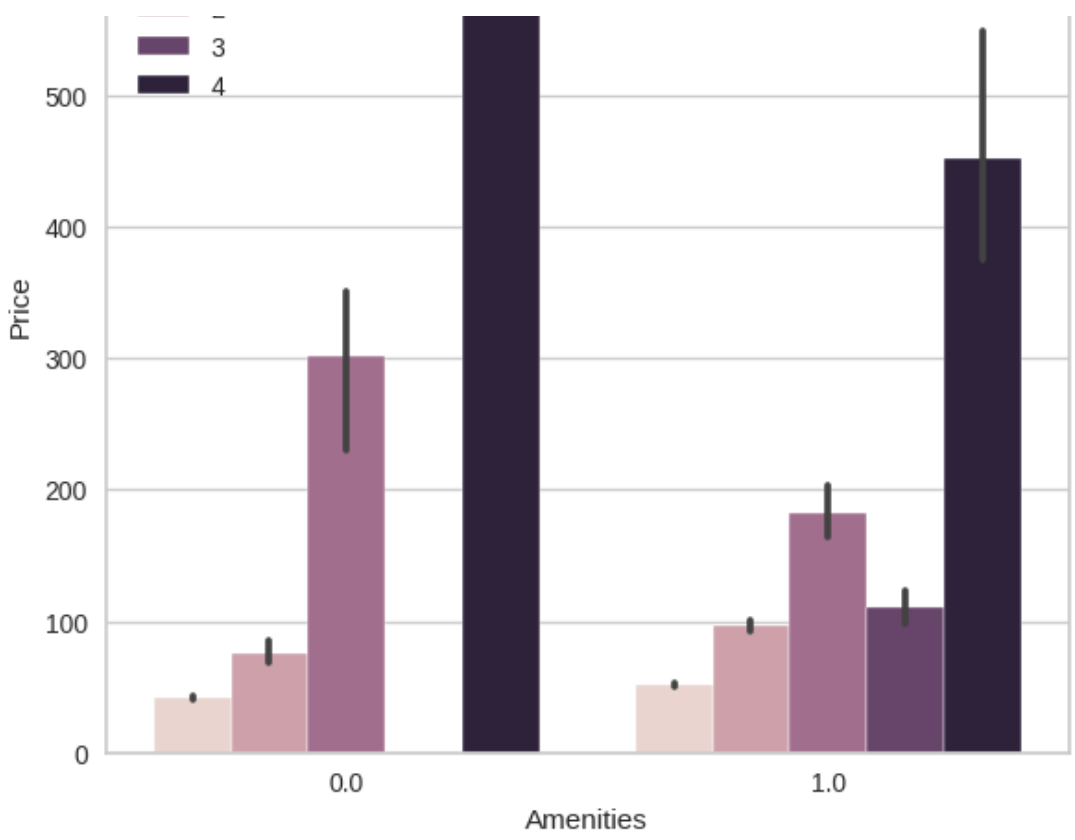




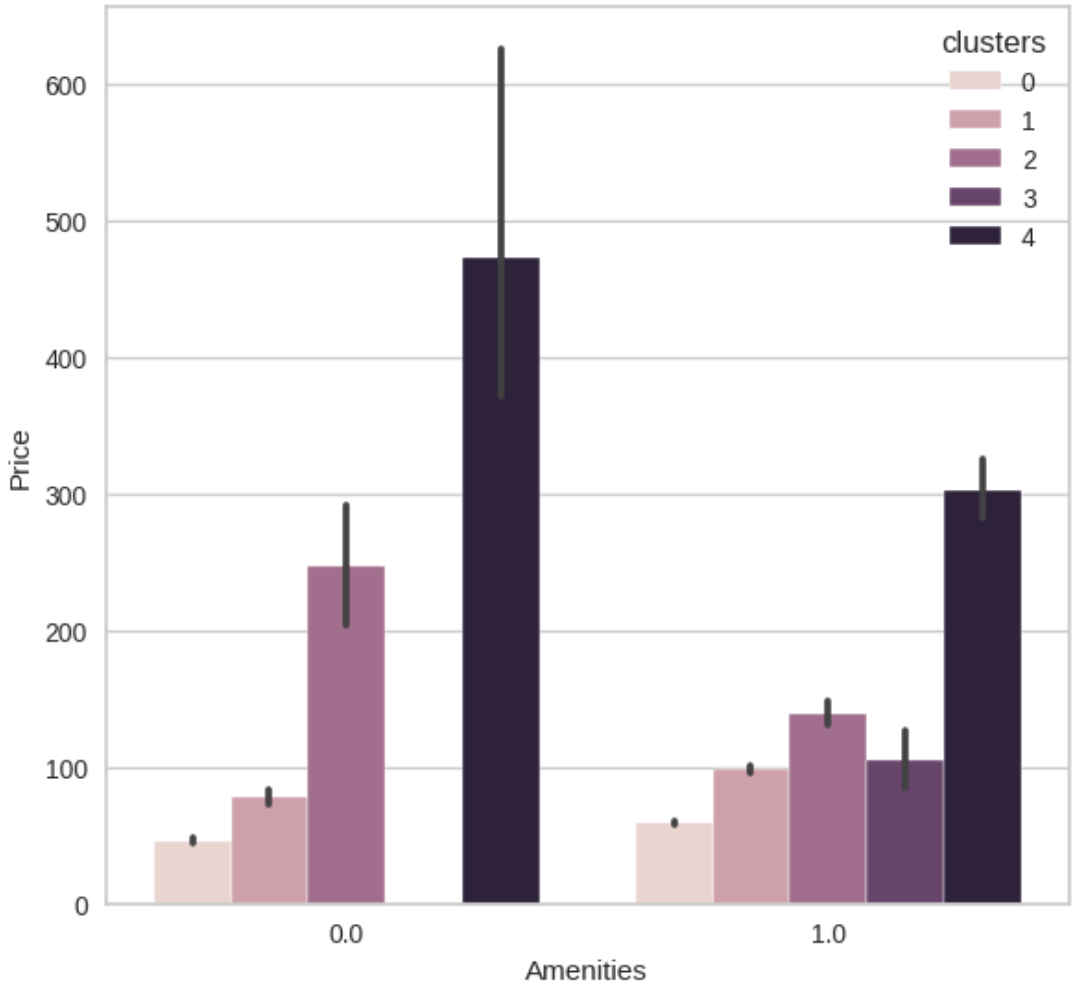
price change with Amenities in Bangalore



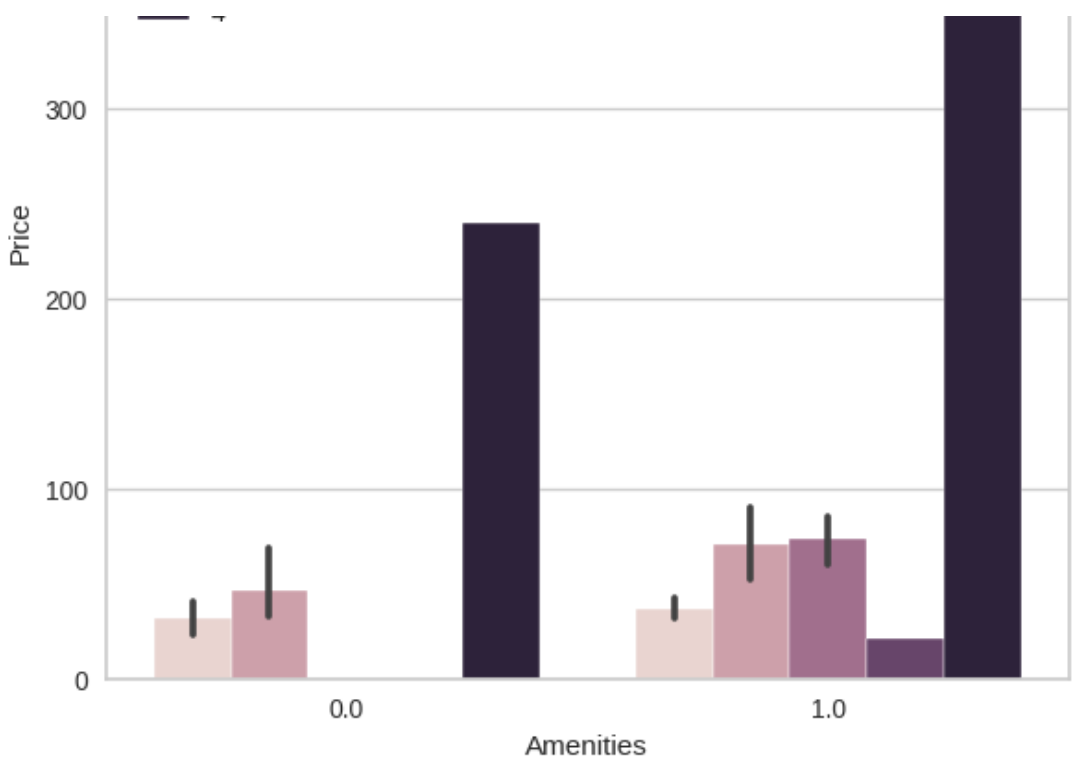
price change with Amenities in Chennai

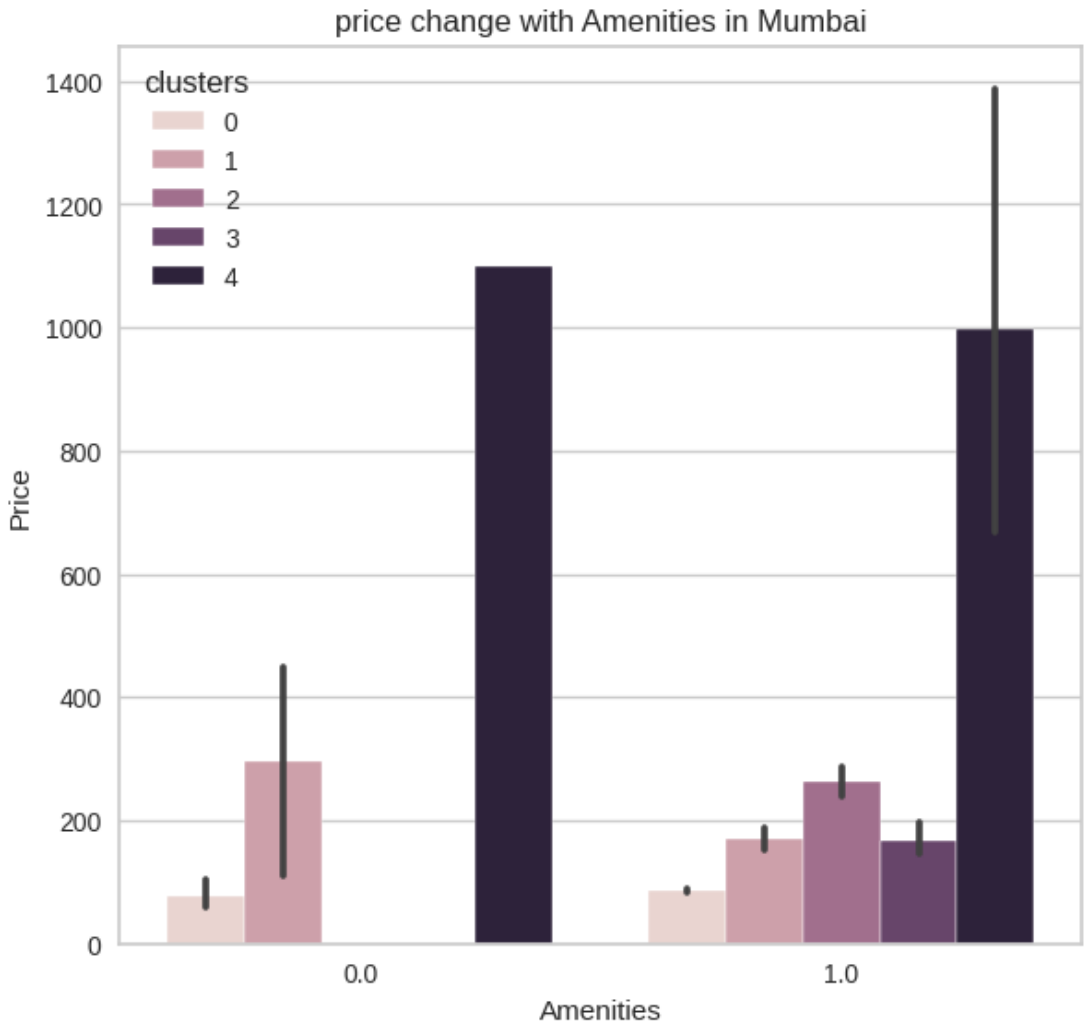


price change with Amenities in Hyderabad



price change with Amenities in Kolkata



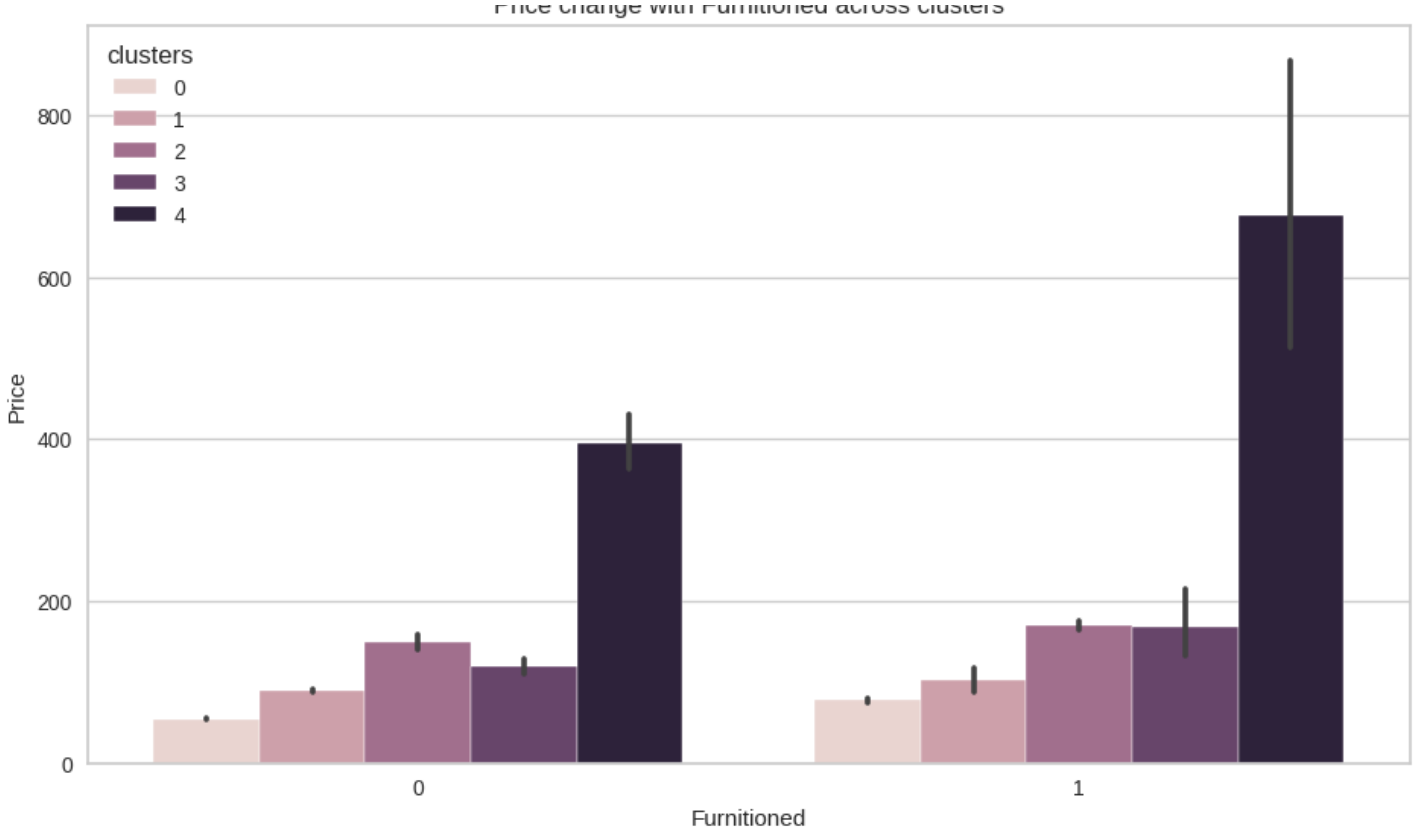


Furnitioned

In [154]:

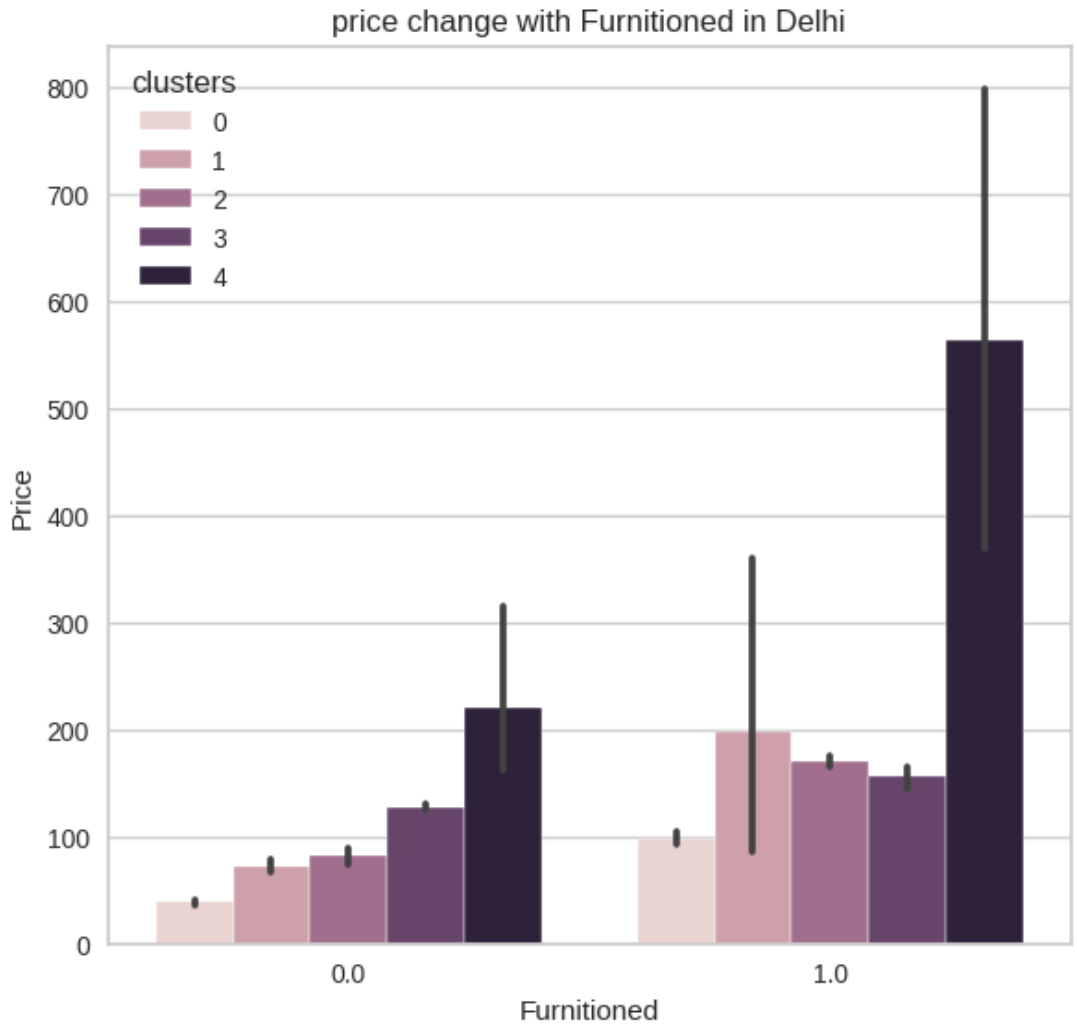
```
plt.figure(figsize=(10, 6))
sns.barplot(x=df['Furnitioned'], y=df['Price'], hue=df['clusters'])
plt.title('Price change with Furnitioned across clusters')
plt.show()
```

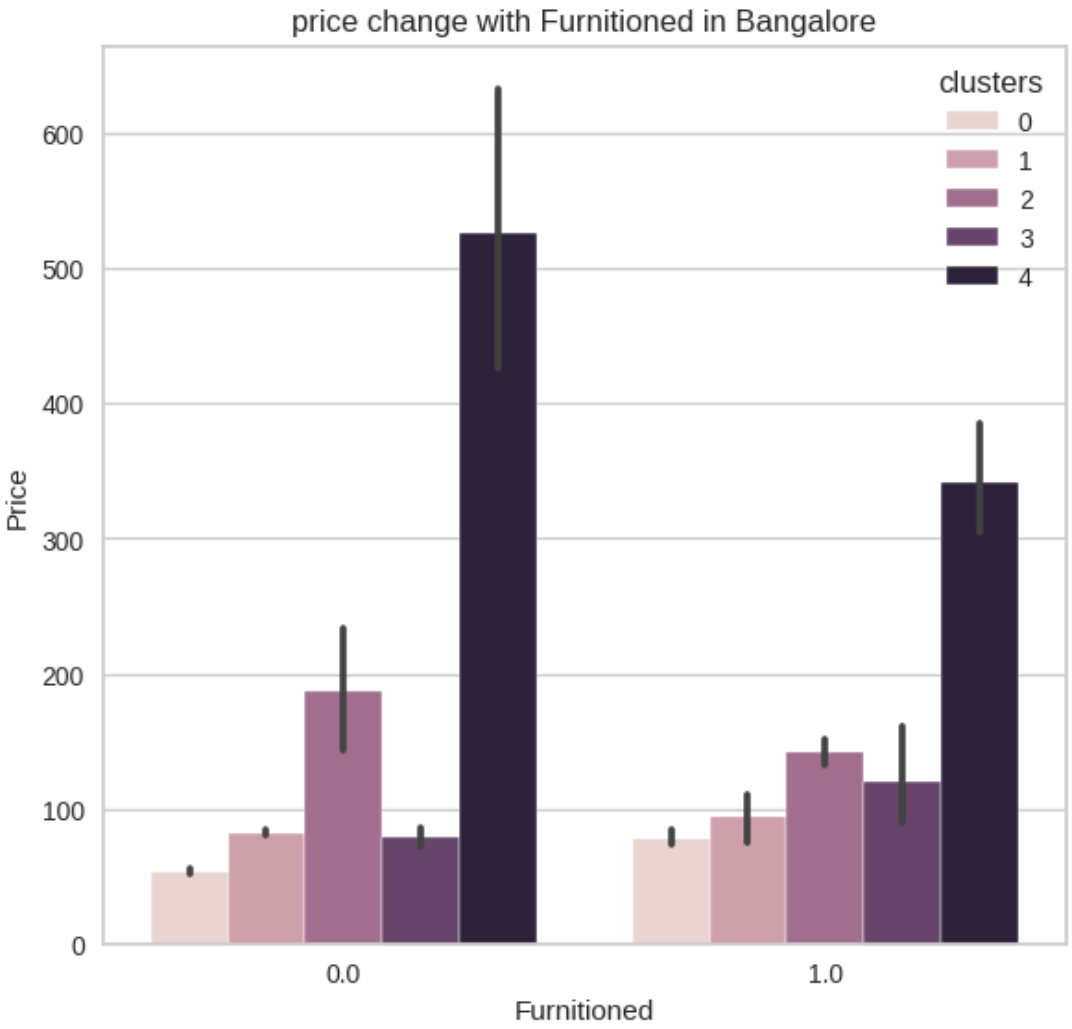
Price change with Furnitioned across clusters

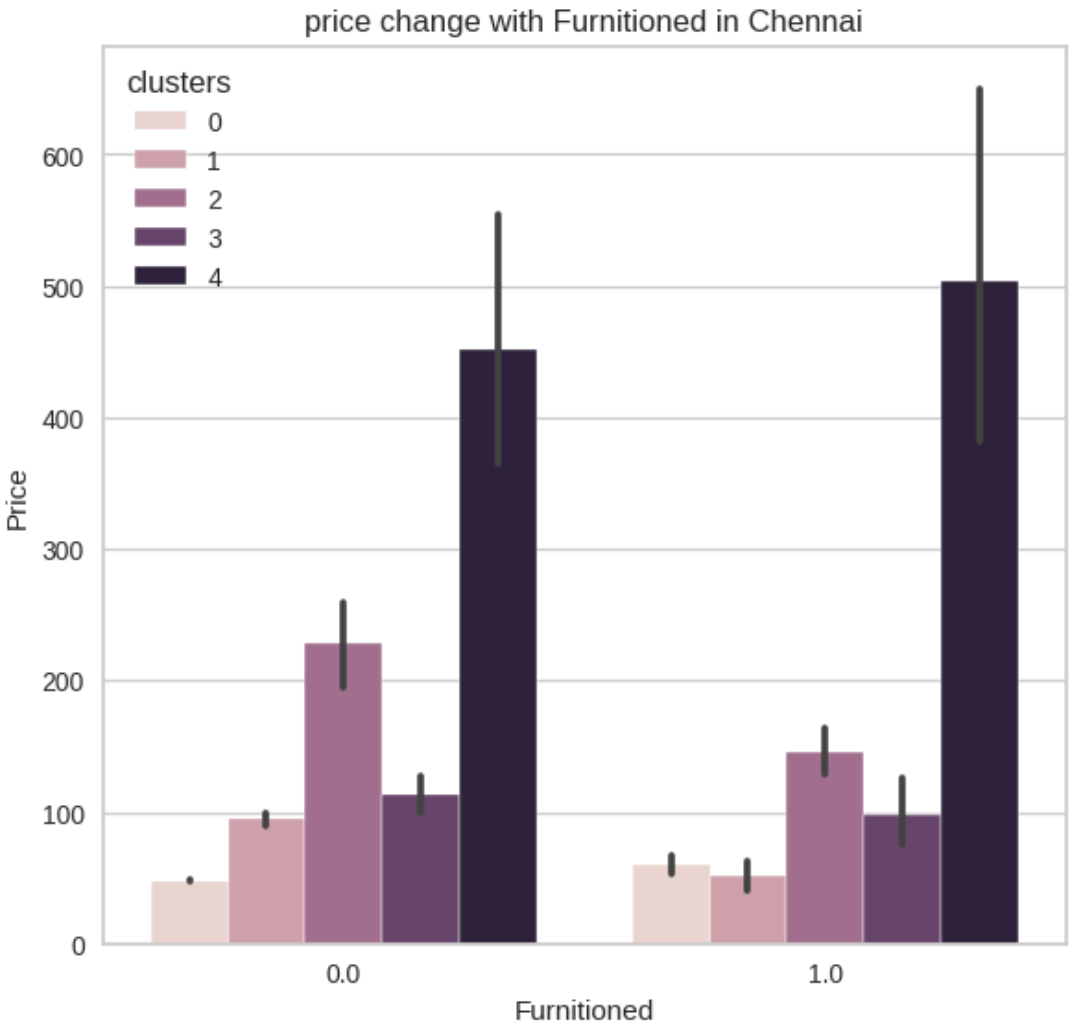


In [155]:

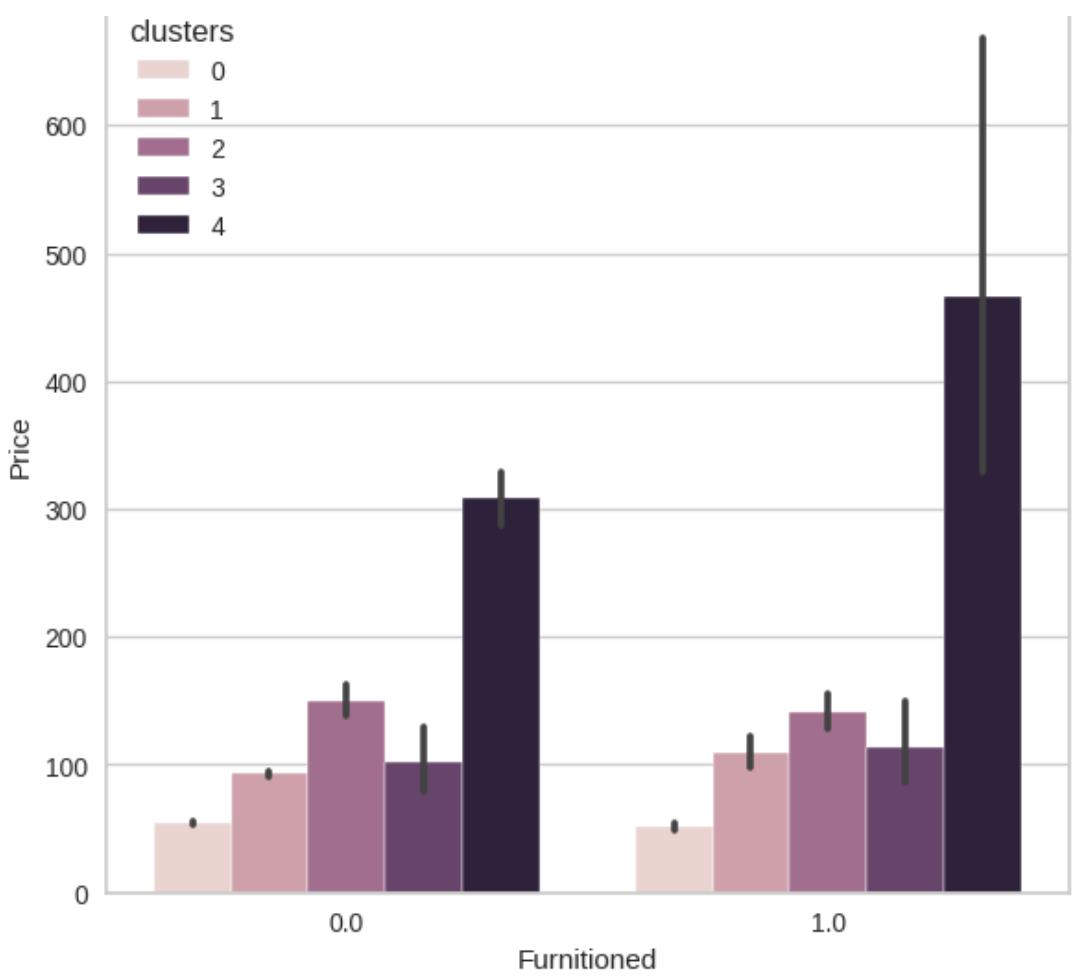
```
for i in city:
  tmpdf = df[df['City'] == i]
  plt.figure(figsize=(6, 6))
  sns.barplot(x='Furnitioned', y='Price', hue=df['clusters'], data=tmpdf)
  plt.title(f"price change with Furnitioned in {i}")
  plt.show()
```



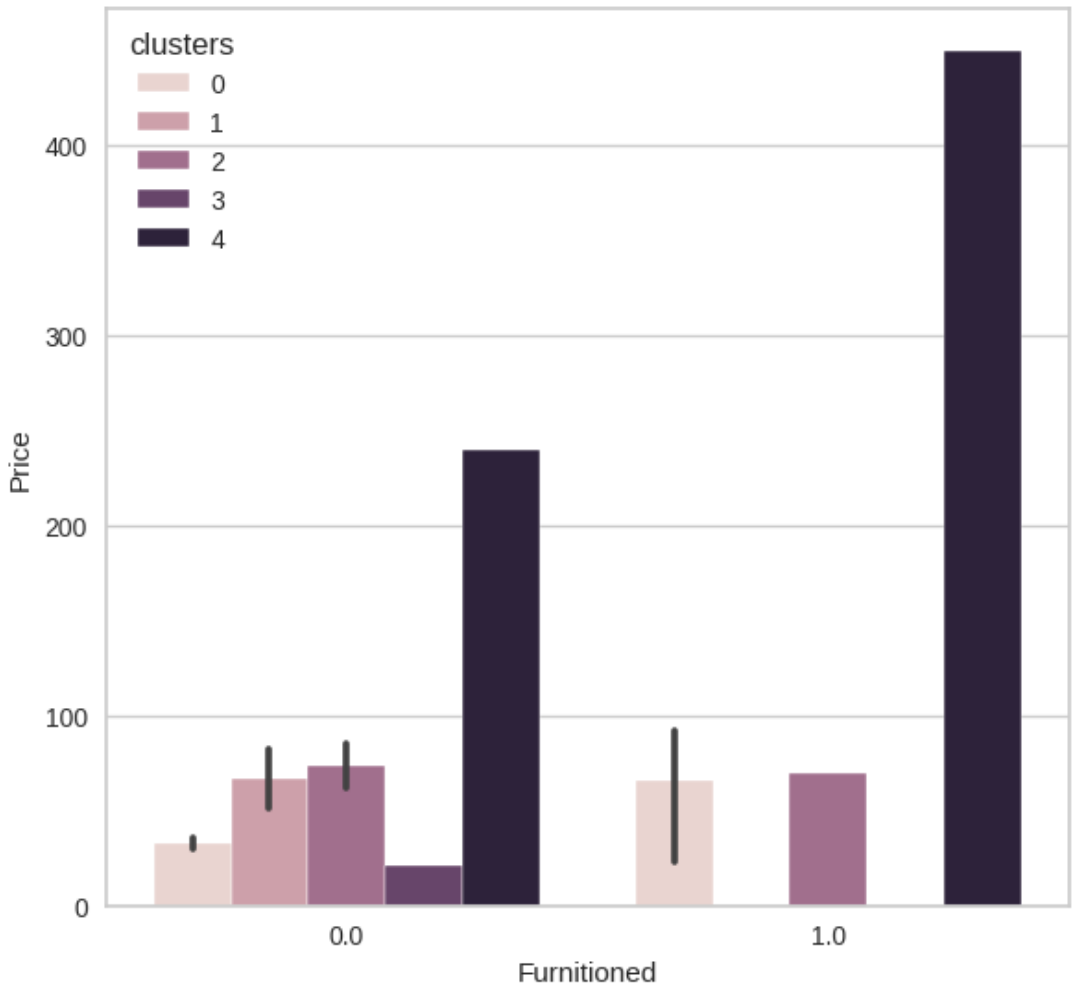

price change with Furnitioned in Bangalore
clusters
0
1
2
3
4
600
500
400
300
200
100
0
Price
0.0
1.0
Furnitioned


price change with Furnitioned in Chennai
clusters
0
1
2
3
4
600
500
400
300
200
100
0
Price
0.0
1.0
Furnitioned

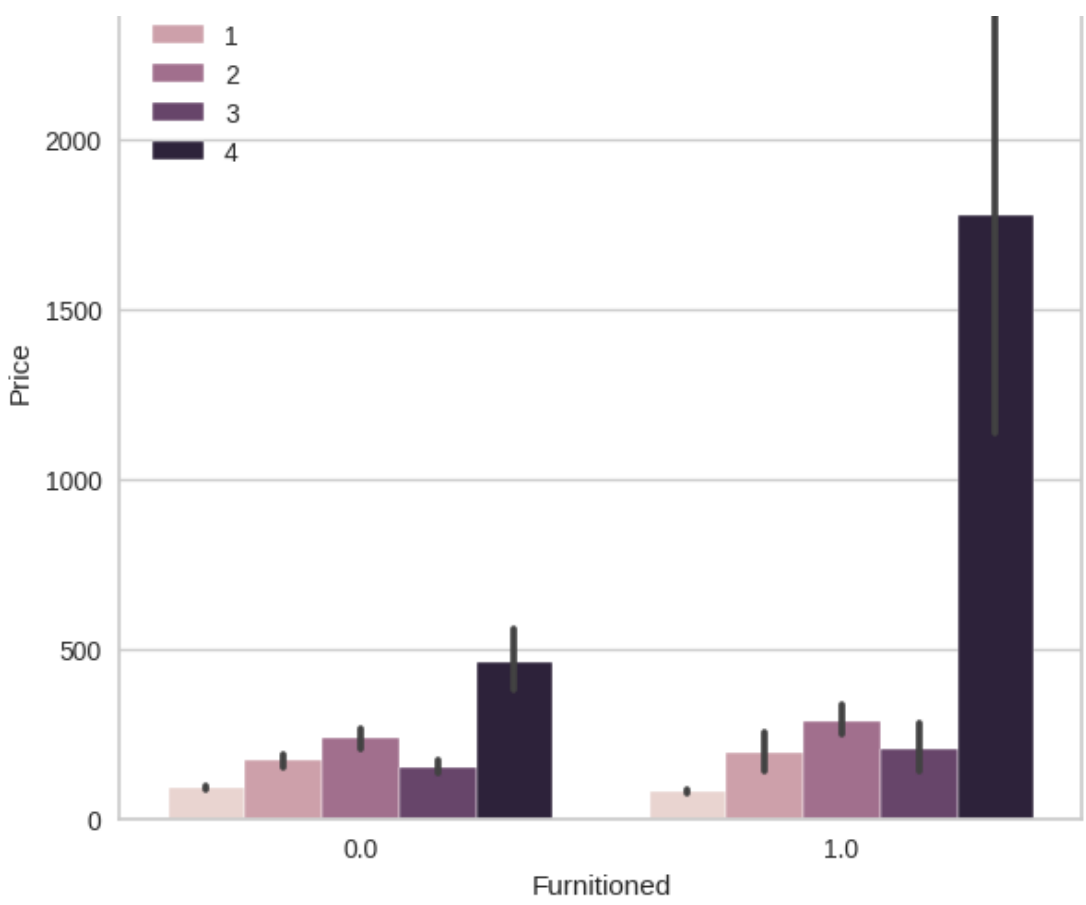

price change with Furnitioned in Hyderabad
700



price change with Furnitioned in Kolkata



price change with Furnitioned in Mumbai



In [ ]: