In [2]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_csv('pizza_sales.csv')
```

In [4]:
```
df
```

Out[4]:

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_p
0	1.0	1.0	hawaiian_m	1.0	1/1/2015	11:38:36	13.25	13
1	2.0	2.0	classic_dlx_m	1.0	1/1/2015	11:57:40	16.00	16
2	3.0	2.0	five_cheese_l	1.0	1/1/2015	11:57:40	18.50	18
3	4.0	2.0	ital_supr_l	1.0	1/1/2015	11:57:40	20.75	20
4	5.0	2.0	mexicana_m	1.0	1/1/2015	11:57:40	16.00	16
...	...	...	...	...	...	...	...	
48615	48616.0	21348.0	ckn_alfredo_m	1.0	31-12-2015	21:23:10	16.75	16
48616	48617.0	21348.0	four_cheese_l	1.0	31-12-2015	21:23:10	17.95	17
48617	48618.0	21348.0	napolitana_s	1.0	31-12-2015	21:23:10	12.00	12
48618	48619.0	21349.0	mexicana_l	1.0	31-12-2015	22:09:54	20.25	20
48619	48620.0	21350.0	bbq_ckn_s	1.0	31-12-2015	23:02:05	12.75	12

48620 rows × 12 columns

In [12]:

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

In [13]:

```python
features = ['quantity', 'unit_price', 'total_price']
df_selected = df[features].dropna()

```

In [14]:

```python
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_selected)
```

In [15]:

```python
df_scaled
```

Out[15]:

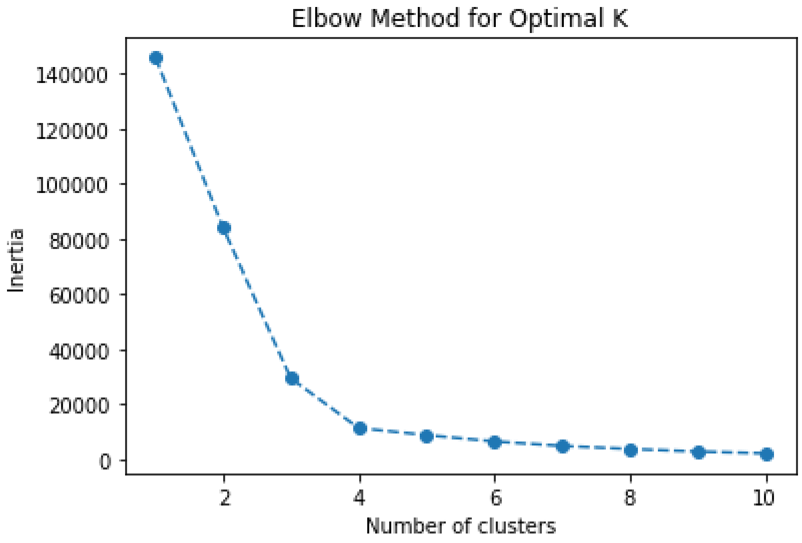
```
array([[-0.13714123, -0.89573558, -0.80486599],
       [-0.13714123, -0.13643454, -0.185127  ],
       [-0.13714123,  0.55383914,  0.37827207],
       ...,
       [-0.13714123, -1.24087242, -1.08656553],
       [-0.13714123,  1.03703071,  0.77265143],
       [-0.13714123, -1.03379031, -0.91754581]])
```

In [17]:

```python
inertia = []
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters = k, random_state = 42, n_init = 10)
    kmeans.fit(df_scaled)
    inertia.append(kmeans.inertia_)
```

In [22]:

```python
plt.figure(figsize = (6, 4))
plt.plot(k_values, inertia, marker = 'o', linestyle = '--')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
plt.show()
```



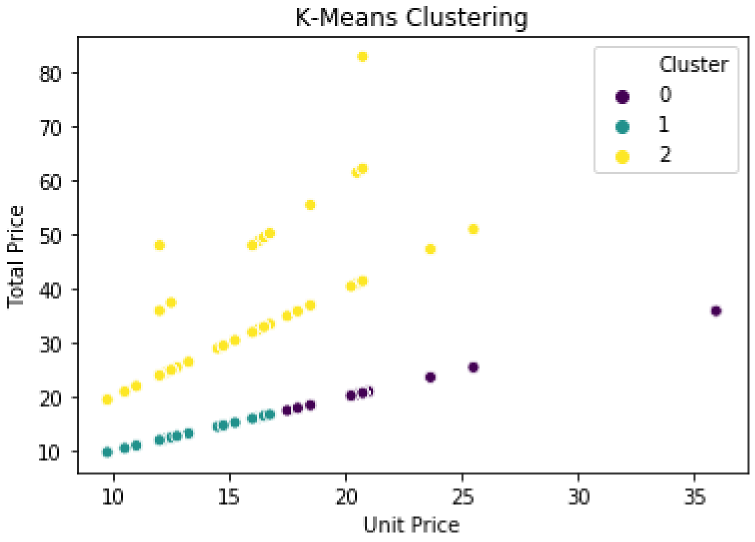
In [26]:

```
kmeans = KMeans(n_clusters = 3, random_state = 42, n_init = 10)
df['Cluster'] = kmeans.fit_predict(df_scaled)

```

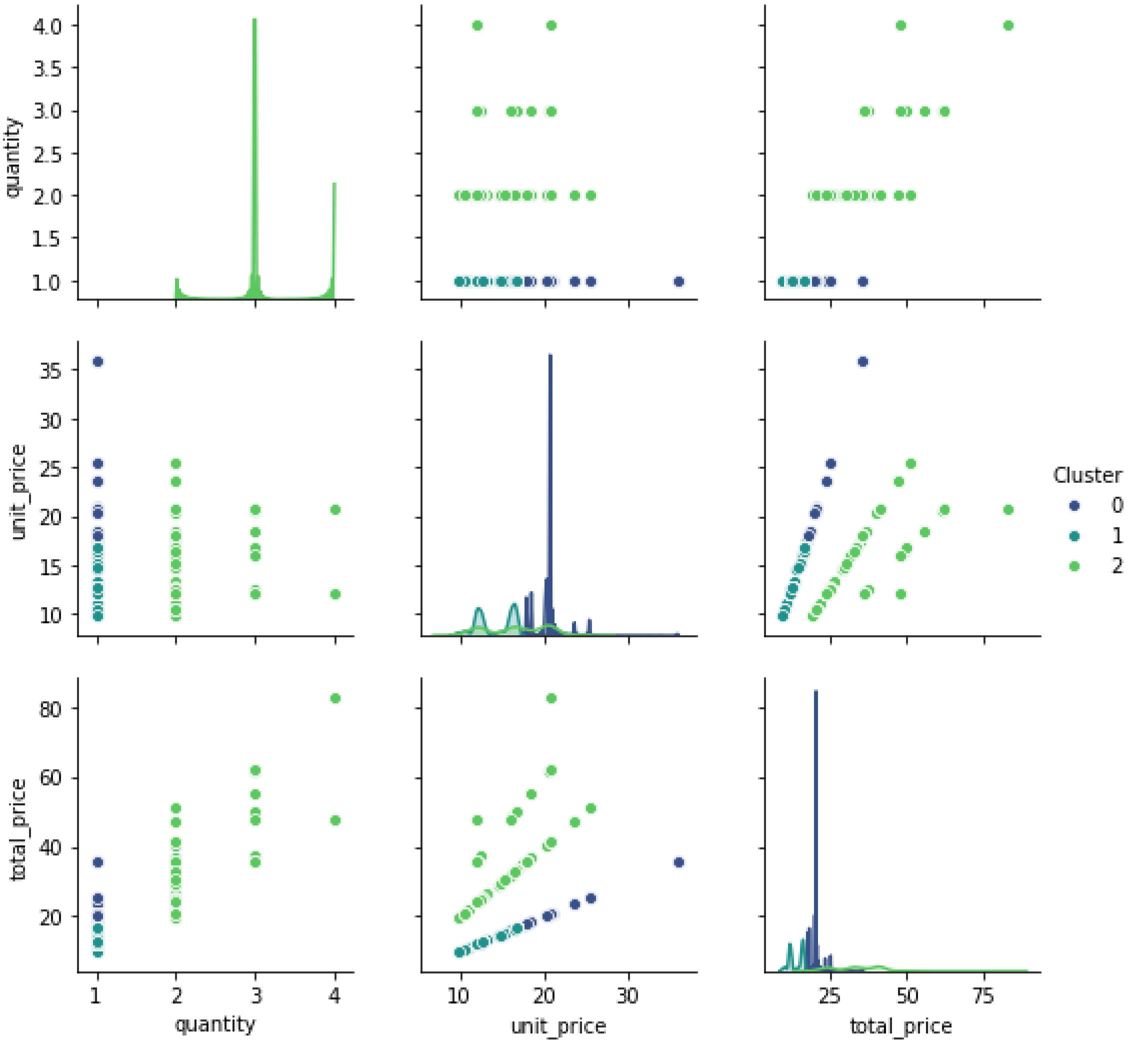
In [27]:

```
plt.figure(figsize = (6,4))
sns.scatterplot(x = df['unit_price'], y = df['total_price'], hue = df['
plt.xlabel('Unit Price')
plt.ylabel('Total Price')
plt.title('K-Means Clustering')
plt.show()
```



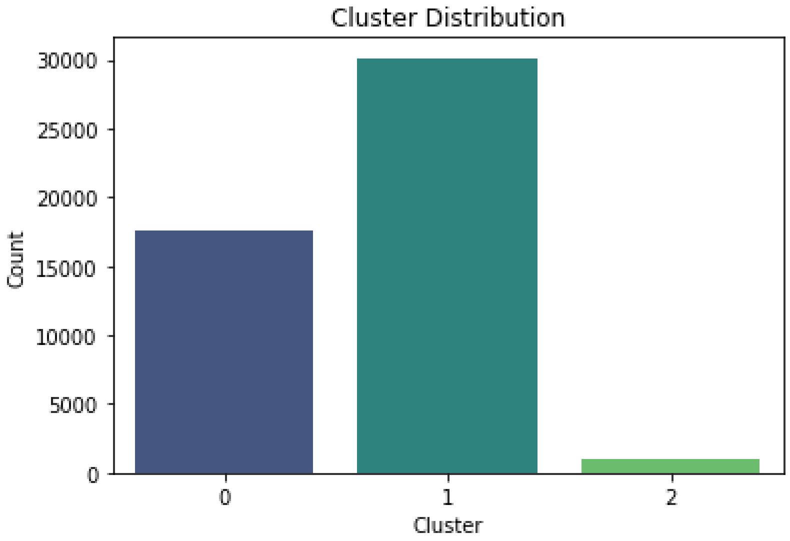
In [29]:

```
sns.pairplot(df_selected.assign(Cluster = df['Cluster']), hue = 'Cluste
plt.show()
```



In [30]:

```
plt.figure(figsize = (6, 4))
sns.countplot(x = df['Cluster'], palette = 'viridis')
plt.xlabel('Cluster')
plt.ylabel('Count')
plt.title('Cluster Distribution')
plt.show()
```



In [ ]: