In [1]:

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

In [2]:

```
#reading the csv file
df = pd.read_csv(r"C:\Users\Administrator\Downloads\Assignment - Junior Data Analyst.csv"
)
```

In [3]:

```
#finding the shape
df.shape
```

Out[3]:

```
(984, 10)
```

In [4]:

```
#knowing total columns
df.columns
```

Out[4]:

```
Index(['battery', 'camera', 'display', 'memory', 'name', 'price', 'processor',
       'rating', 'reviews', 'warranty'],
      dtype='object')
```

In [5]:

```
#studing the sample
df.head()
```

Out[5]:

	battery	camera	display	memory	name	price	processor	rating	reviews	warranty
0	5000 mAh Battery	12MP + 2MP | 8MP Front Camera	15.8 cm (6.22 inch) HD+ Display	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	Redmi 8 (Ruby Red, 64 GB)	9999	Qualcomm Snapdragon 439 Processor	4.4	55,078 Reviews	Brand Warranty of 1 Year Available for Mobile ...
1	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 64 GB ROM	Realme 5i (Aqua Blue, 64 GB)	10999	Qualcomm Snapdragon 665 2 GHz Processor	4.5	20,062 Reviews	Sunrise Design
2	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Realme 5i (Aqua Blue, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20,062 Reviews	Sunrise Design
3	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Realme 5i (Forest Green, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20,062 Reviews	Sunrise Design
4	4000 mAh Battery	13MP + 2MP | 5MP Front Camera	15.49 cm (6.1 inch) HD+ Display	3 GB RAM | 32 GB ROM | Expandable Upto 256 GB	Realme C2 (Diamond Blue, 32 GB)	7499	MediaTek P22 Octa Core 2.0 GHz Processor	4.4	10,091 Reviews	Dual Nano SIM slots and Memory Card Slot

In [6]:

```
#finding total null values
df.isna().sum()
```

Out[6]:

```
battery         0
camera          0
display         0
memory          0
name            0
price           0
processor       1
rating         13
reviews        13
warranty      148
dtype: int64
```

In [7]:

```
#deleting duplicates values
df.drop_duplicates()
```

Out[7]:

	battery	camera	display	memory	name	price	processor	rating	reviews	warranty
0	5000 mAh Battery	12MP + 2MP | 8MP Front Camera	15.8 cm (6.22 inch) HD+ Display	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	Redmi 8 (Ruby Red, 64 GB)	9999	Qualcomm Snapdragon 439 Processor	4.4	55,078 Reviews	Brand Warranty of 1 Year Available for Mobile ...
1	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 64 GB ROM	Realme 5i (Aqua Blue, 64 GB)	10999	Qualcomm Snapdragon 665 2 GHz Processor	4.5	20,062 Reviews	Sunrise Design
2	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Realme 5i (Aqua Blue, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20,062 Reviews	Sunrise Design
3	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Realme 5i (Forest Green, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20,062 Reviews	Sunrise Design
4	4000 mAh Battery	13MP + 2MP | 5MP Front Camera	15.49 cm (6.1 inch) HD+ Display	3 GB RAM | 32 GB ROM | Expandable Upto 256 GB	Realme C2 (Diamond Blue, 32 GB)	7499	MediaTek P22 Octa Core 2.0 GHz Processor	4.4	10,091 Reviews	Dual Nano SIM slots and Memory Card Slot
...	...	...	...	...	...	...	...	...	...	...
979	2000 mAh Battery	5MP Rear Camera | 2MP Front Camera	12.7 cm (5 inch) FWVGA Display	1 GB RAM | 8 GB ROM | Expandable Upto 32 MB	Micromax Bharat 4 (Black, 8 GB)	3590	12 Months Brand Warranty	3.8	105 Reviews	NaN
980	2680 mAh Li-Ion Battery	13MP Rear Camera | 5MP Front Camera	13.21 cm (5.2 inch) Full HD Display	3 GB RAM | 32 GB ROM	Nextbit Robin (Ember, 32 GB)	19999	Qualcomm Snapdragon 808 MSM8992 Processor	4.0	516 Reviews	Brand Warranty of 1 Year
981	4550 mAh Battery	13MP + 5MP | 20MP Front Camera	15.24 cm (6 inch) Full HD Display	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	Gionee A1 Plus (Mocha Gold, 64 GB)	10499	Helio P25 MT 6757CD Processor	4.1	710 Reviews	Brand Warranty of 1 Year Available for Mobile ...
982	2100 mAh Li-Ion Battery	8MP Rear Camera | 2MP Front Camera	12.7 cm (5 inch) HD Display	1 GB RAM | 8 GB ROM | Expandable Upto 32 GB	XOLO Omega 5.0 (Black, 8 GB)	8990	MTK 6592M Processor	3.8	81 Reviews	1 Year Manufacturer Warranty
					Samsung					Brand

	battery	camera	display	memory	name	price	processor	rating	reviews	warranty
983	3000 mAh Battery	12MP Rear Camera + 8MP Front Camera	14.73 cm (5.8 inch) Quad HD+ Display	4 GB RAM | 256 GB ROM | Expandable Upto 400 GB	Galaxy S9 (Midnight Black, 256 GB)	65900	Exynos 9810 Processor	4.4	2,351 Reviews	Warranty of 1 Year Available for Mobile ...

960 rows × 10 columns

In [8]:

```
#removing nan values for better analysis
df.dropna(inplace = True)
```

In [9]:

```
#understanding all columns datatypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 828 entries, 0 to 983
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   battery    828 non-null    object 
 1   camera     828 non-null    object 
 2   display    828 non-null    object 
 3   memory     828 non-null    object 
 4   name       828 non-null    object 
 5   price      828 non-null    int64  
 6   processor  828 non-null    object 
 7   rating     828 non-null    float64
 8   reviews    828 non-null    object 
 9   warranty   828 non-null    object 
dtypes: float64(1), int64(1), object(8)
memory usage: 71.2+ KB
```

In [10]:

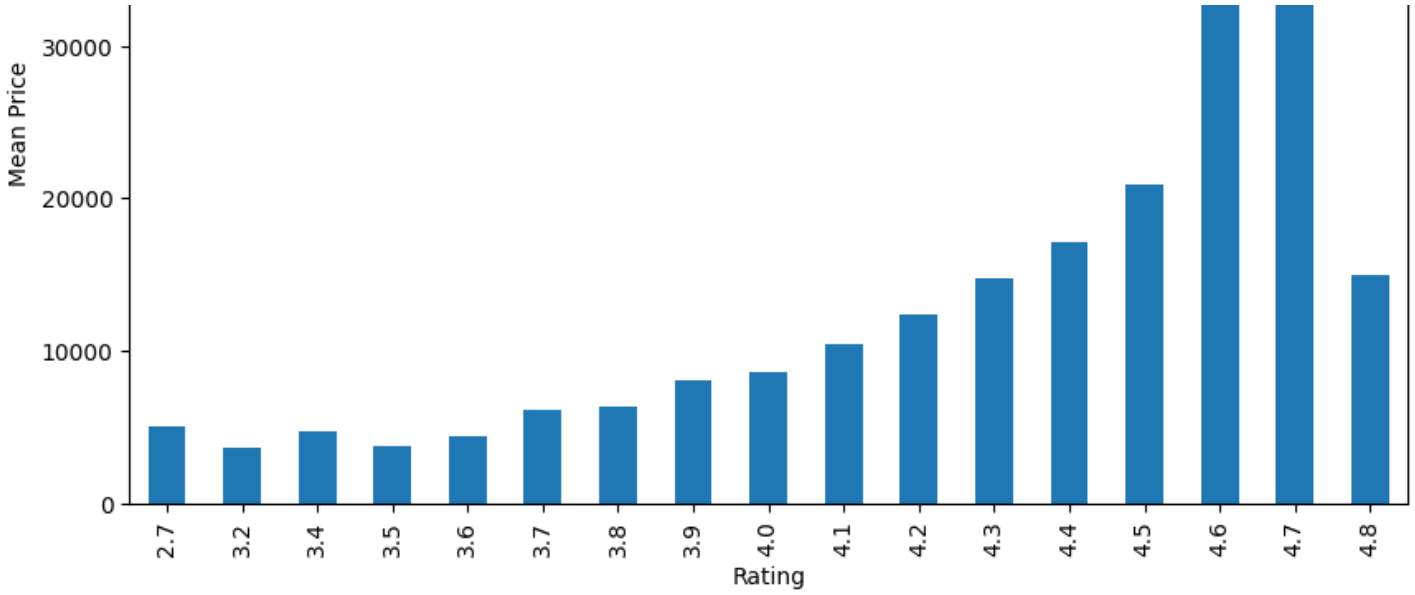
```
#finding total unique names
df['name'].nunique()
```

Out[10]:

```
730
```

In [11]:

```
#grouping ratings based on price avg

df_group = df.groupby(by='rating')['price'].mean().round(0)

# Plot the result
plt.figure(figsize=(10, 6))
df_group.plot(kind='bar')

# Set plot title and labels
plt.title('Mean Price by Rating')
plt.xlabel('Rating')
plt.ylabel('Mean Price')

# Show the plot
plt.show()
```



Higher-rated phones are generally more expensive, with a substantial increase in price around the 4.5-4.7 rating range. Lower-rated phones tend to be cheaper, justifying the idea that premium phones are often rated higher. There are some exceptions, indicating that rating is not the only factor influencing price, but it's a strong indicator.

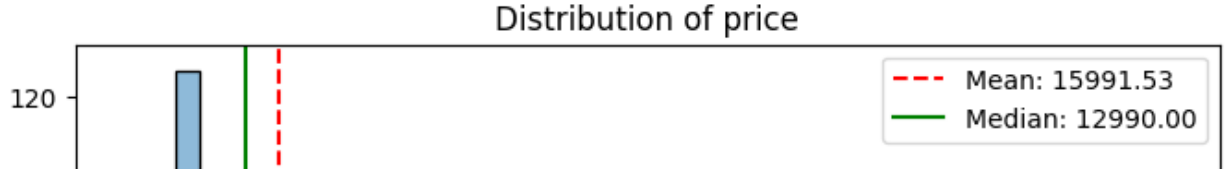
Univariate anlaysis of all columns

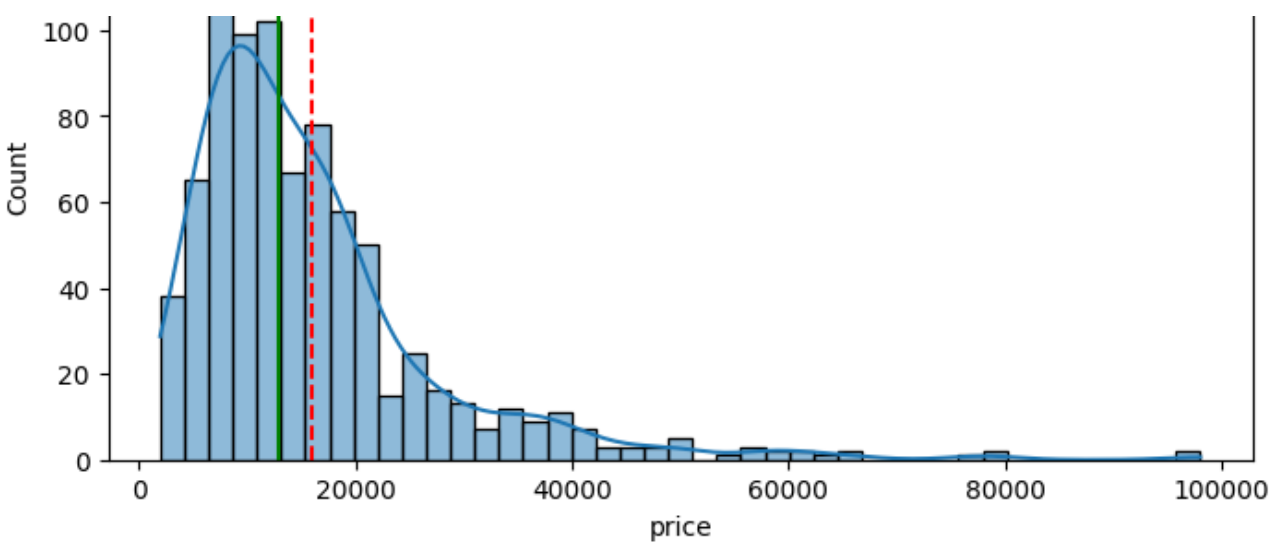
In [12]:

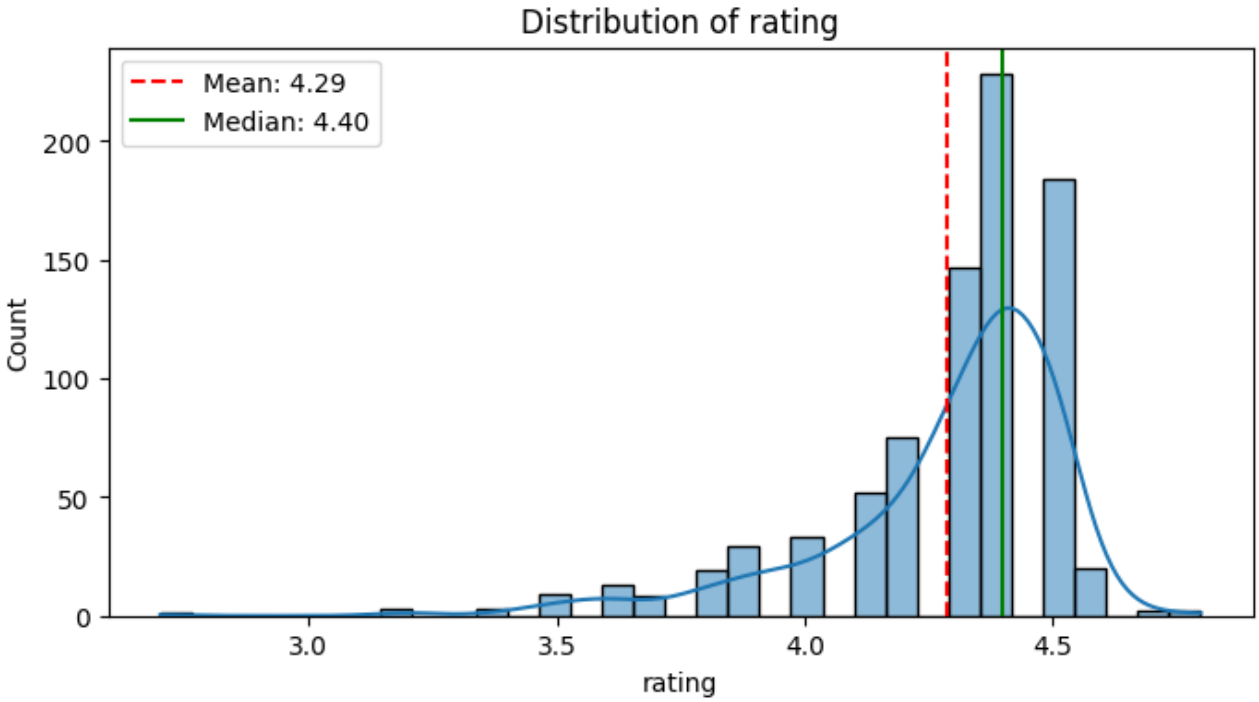
```
# so the data in the column reviews have int and str both values(object) but that str wil
l cause some issue while performing the analysis.
# Thats why, I am excluding the string from the data to make it a numerical data for bett
er analysis.


df['reviews'] = df['reviews'].astype('str')
df['reviews'] = df['reviews'].apply(lambda row: row[:row.find(' ')])
df['reviews'] = df['reviews'].str.replace(',','')
df['reviews'] = pd.to_numeric(df['reviews'], errors= 'coerce')
```

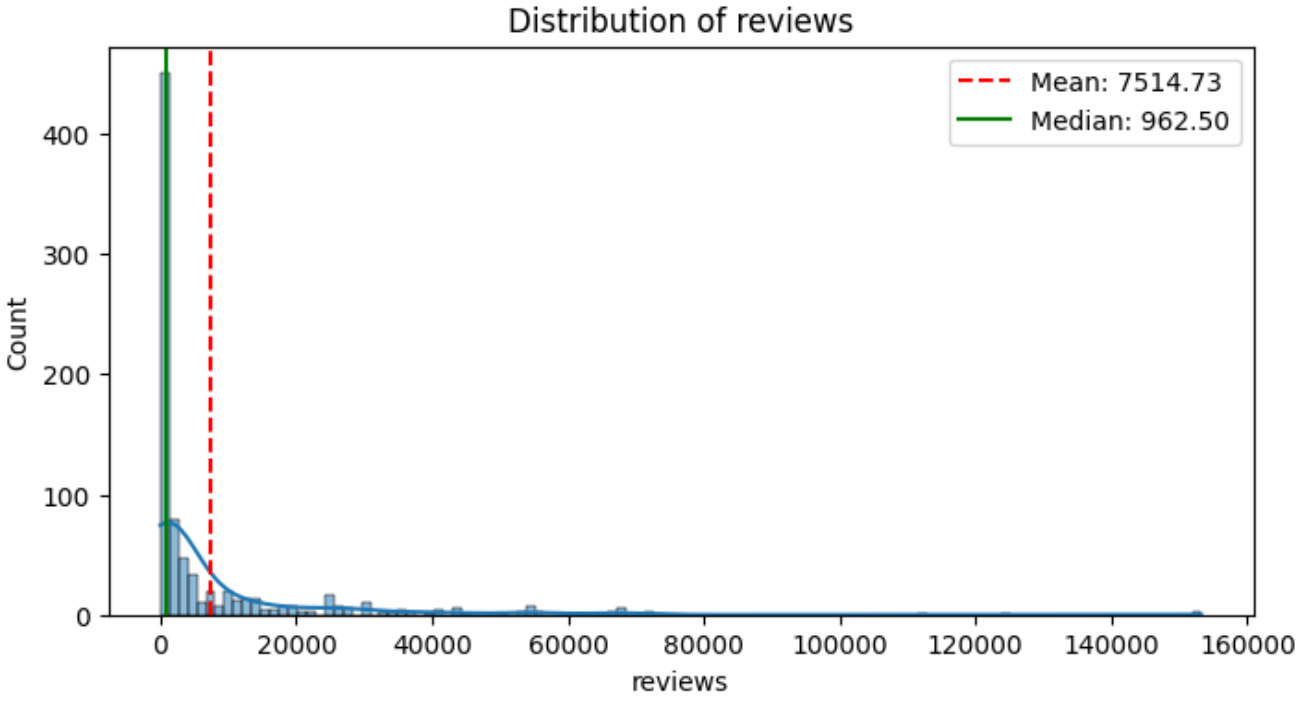
In [13]:

```
#Numerical columns plotting


numerical_cols = ['price', 'rating', 'reviews']

for col in numerical_cols:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col].dropna(), kde=True)
    plt.title(f'Distribution of {col}')

    # Add descriptive statistics to the plot
    mean_val = df[col].mean()
    median_val = df[col].median()
    std_val = df[col].std()

    plt.axvline(mean_val, color='r', linestyle='--', label=f'Mean: {mean_val:.2f}')
    plt.axvline(median_val, color='g', linestyle='-', label=f'Median: {median_val:.2f}')
    plt.legend()

    plt.show()
```









Plot 1st: Overall, the data shows a concentration of lower-priced items with a few high-priced outliers, suggesting that the average price may not be a representative measure of the typical price.

Plot 2nd: Overall, the data shows a concentration of higher ratings with a few lower-rated outliers, suggesting that the average rating may not be a representative measure of the typical rating.

Plot 3rd: Overall, the data shows a concentration of items with a small number of reviews with a few highly reviewed outliers, suggesting that the average number of reviews may not be a representative measure of the typical number of reviews.

In [14]:

```
#studying Categroical columns

categorical_cols = ['battery', 'camera', 'display', 'memory', 'processor', 'warranty']
df[categorical_cols]
```

Out[14]:

	battery	camera	display	memory	processor	warranty
0	5000 mAh Battery	12MP + 2MP | 8MP Front Camera	15.8 cm (6.22 inch) HD+ Display	4 GB RAM | 64 GB ROM | Expandable Upto 512 GB	Qualcomm Snapdragon 439 Processor	Brand Warranty of 1 Year Available for Mobile ...
1	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 64 GB ROM	Qualcomm Snapdragon 665 2 GHz Processor	Sunrise Design
2	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Qualcomm Snapdragon 665 (2 GHz) Processor	Sunrise Design
3	5000 mAh Battery	12MP + 8MP + 2MP + 2MP | 8MP Front Camera	16.56 cm (6.52 inch) HD+ Display	4 GB RAM | 128 GB ROM	Qualcomm Snapdragon 665 (2 GHz) Processor	Sunrise Design
4	4000 mAh Battery	13MP + 2MP | 5MP Front Camera	15.49 cm (6.1 inch) HD+ Display	3 GB RAM | 32 GB ROM | Expandable Upto 256 GB	MediaTek P22 Octa Core 2.0 GHz Processor	Dual Nano SIM slots and Memory Card Slot
...	...	...	...	...	...	...
978	2300 mAh Li-Ion Polymer Battery	8MP Rear Camera | 5MP Front Camera	12.7 cm (5 inch) HD Display	1 GB RAM | 16 GB ROM | Expandable Upto 128 GB	Quad Core 1.3GHz Processor	Brand Warranty of 1 Year
980	2680 mAh Li-Ion Battery	13MP Rear Camera | 5MP Front Camera	13.21 cm (5.2 inch) Full HD Display	3 GB RAM | 32 GB ROM	Qualcomm Snapdragon 808 MSM8992 Processor	Brand Warranty of 1 Year
981	4550 mAh Battery	13MP + 5MP | 20MP Front Camera	15.24 cm (6 inch) Full HD Display	4 GB RAM | 64 GB ROM | Expandable Upto 256 GB	Helio P25 MT 6757CD Processor	Brand Warranty of 1 Year Available for Mobile ...
982	2100 mAh Li-Ion Battery	8MP Rear Camera | 2MP Front Camera	12.7 cm (5 inch) HD Display	1 GB RAM | 8 GB ROM | Expandable Upto 32 GB	MTK 6592M Processor	1 Year Manufacturer Warranty
983	3000 mAh Battery	12MP Rear Camera | 8MP Front Camera	14.73 cm (5.8 inch) Quad HD+ Display	4 GB RAM | 256 GB ROM | Expandable Upto 400 GB	Exynos 9810 Processor	Brand Warranty of 1 Year Available for Mobile ...

828 rows × 6 columns

So, here what I found that the categorical columns have long data which is making the analysis/plot so much messy so I am extracting the main/important key words from data.

In [15]:

```
#extracting the data when the key starts with MP

import re
df['camera'] = df['camera'].apply(lambda x: ' '.join(re.findall(r'\d+MP', x)))
```

In [16]:

```
#extracting the data when the key starts with GB

df['memory'] = df['memory'].apply(lambda x: ' '.join(re.findall(r'\d+ GB', x)))
```

In [17]:

```
df['display'] = df['display'].apply(lambda x: x[:x.find('cm')])
```

In [18]:

```
df['battery'] = df['battery'].apply(lambda x: x[:x.find('mAh')] if 'mAh' in x else x[:x.
find('mAH')] if 'mAH' in x else x)
```

while working on data I found there are too much outliers present out there that will give very negative impact to our analysis and our models. Therefore, it is necessary to tackle the outliers.

Either we can drop that outliers by finding it or make it null as the data itself is very low volumne thats why I am making it null

In [19]:

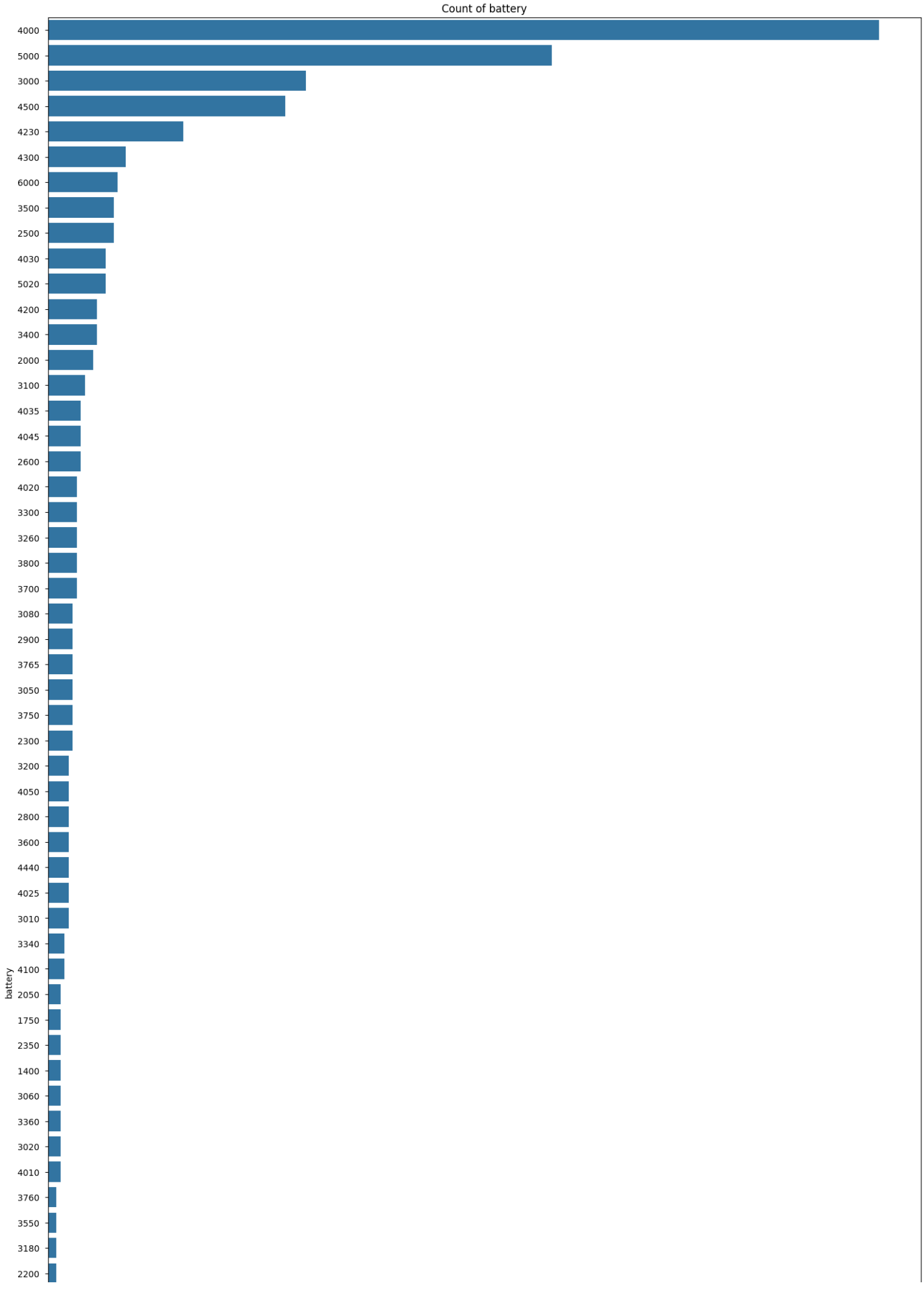
```
#inorder to find outliers I am assigning a keyword. If the data inside finds the keyword
then its fine else its outlier.


df['processor'] = df['processor'].astype('str')

valid_keywords = [ 'Processor']

#MAking a Function to identify if a row is valid or an outlier
def is_valid_warranty(text):
    return any(keyword.lower() in text.lower() for keyword in valid_keywords)

# Apply the function to filter out outliers
df['is_valid'] = df['processor'].apply(is_valid_warranty)

# Separate outliers & valid data
outliers = df[~df['is_valid']]
valid_entries = df[df['is_valid']]

df.loc[~df['is_valid'],'processor']=''
df.drop(columns=['is_valid'],inplace = True)
```

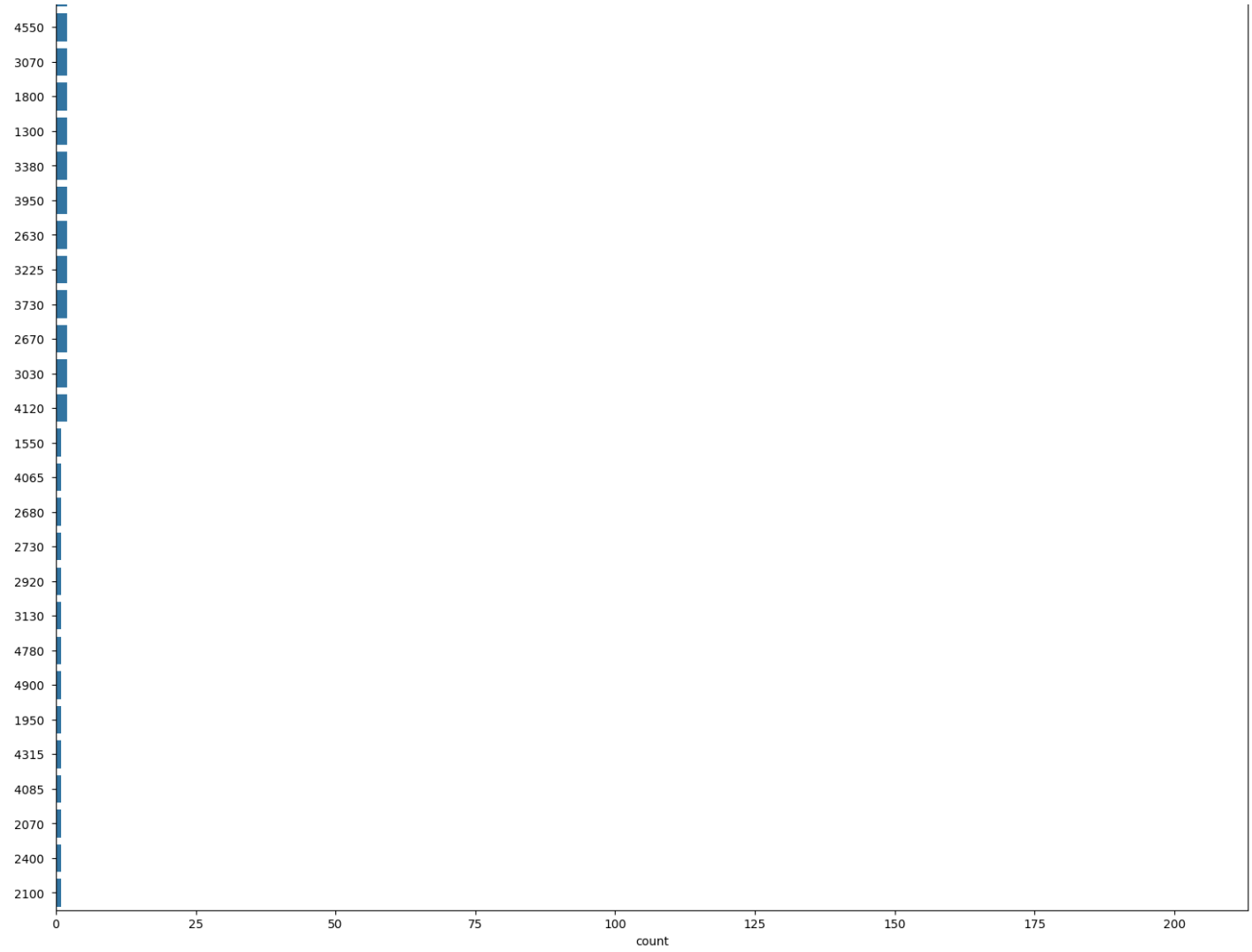
In [20]:

```
#same as above

df['warranty'] = df['warranty'].astype('str')

valid_keywords = ['warranty']

# Function to identify if a row is valid or an outlier
def is_valid_warranty(text):
    return any(keyword.lower() in text.lower() for keyword in valid_keywords)

# Apply the function to filter out outliers
df['is_valid'] = df['warranty'].apply(is_valid_warranty)

# Separate outliers
outliers = df[~df['is_valid']]
valid_entries = df[df['is_valid']]

df.loc[~df['is_valid'],'warranty']=''
df.drop(columns=['is_valid'],inplace = True)
```

In [21]:

```
#battery and its counts

plt.figure(figsize=(18, 40))
sns.countplot(data = df,y='battery', order=df['battery'].value_counts().index)
plt.title(f'Count of battery')
plt.show()
```

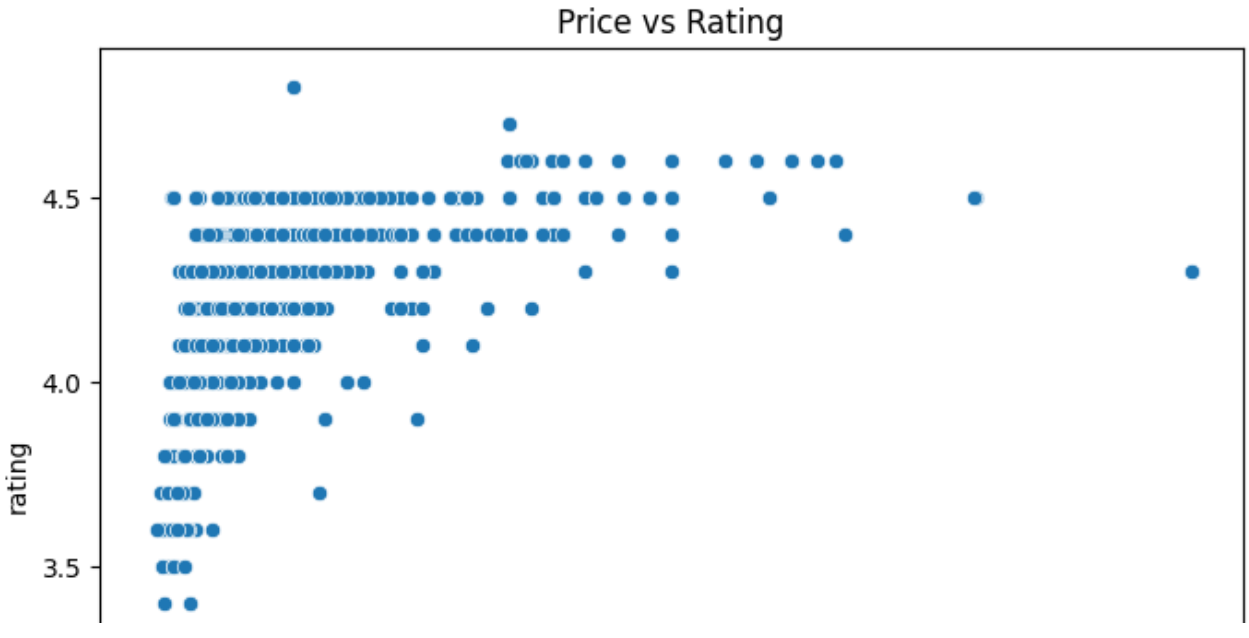


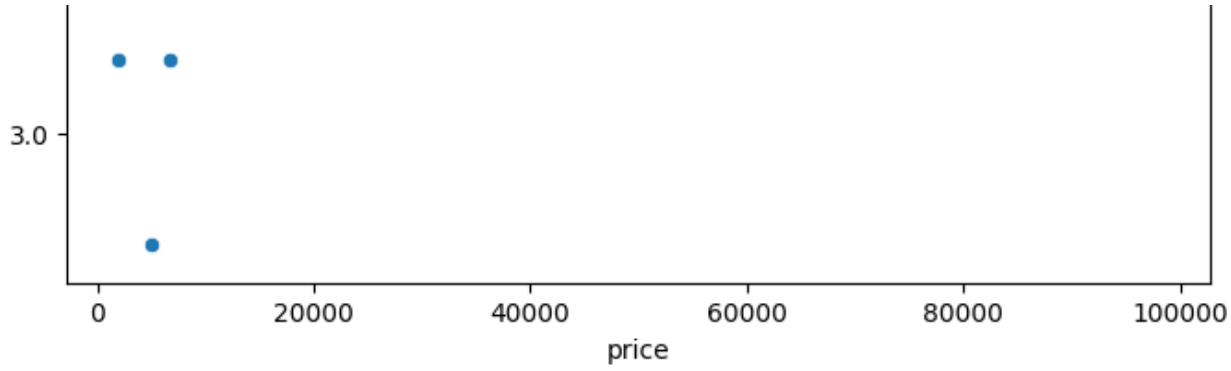


The distribution of battery capacities shows that 4000 mAh and 5000 mAh batteries are the most prevalent in the dataset, with a significantly higher count than other capacities. These two battery sizes dominate the data, suggesting they are either more commonly used in products or appear more frequently in the dataset. Other battery capacities, such as 3000 mAh and 4500 mAh, also appear but in much lower frequencies.

In [22]:

```
# Scatter plot for price vs rating
plt.figure(figsize=(8, 6))
sns.scatterplot(x='price', y='rating', data=df)
plt.title('Price vs Rating')
plt.show()
```

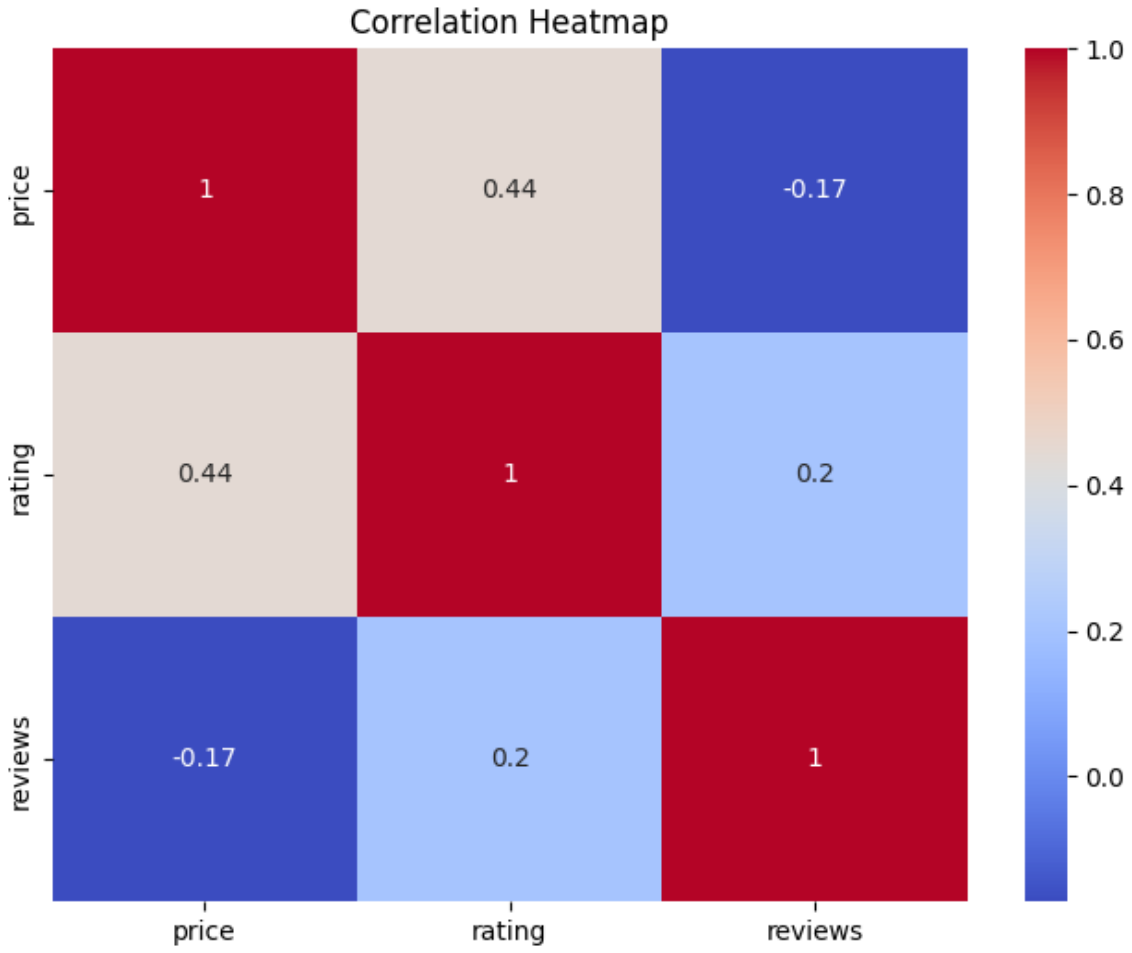




In summary, the data suggests that price does not strongly determine product rating, as high ratings are observed across a wide range of prices.

In [23]:

```
# Correlation heatmap to find relationships between numerical variables

correlation_matrix = df[['price', 'rating', 'reviews']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

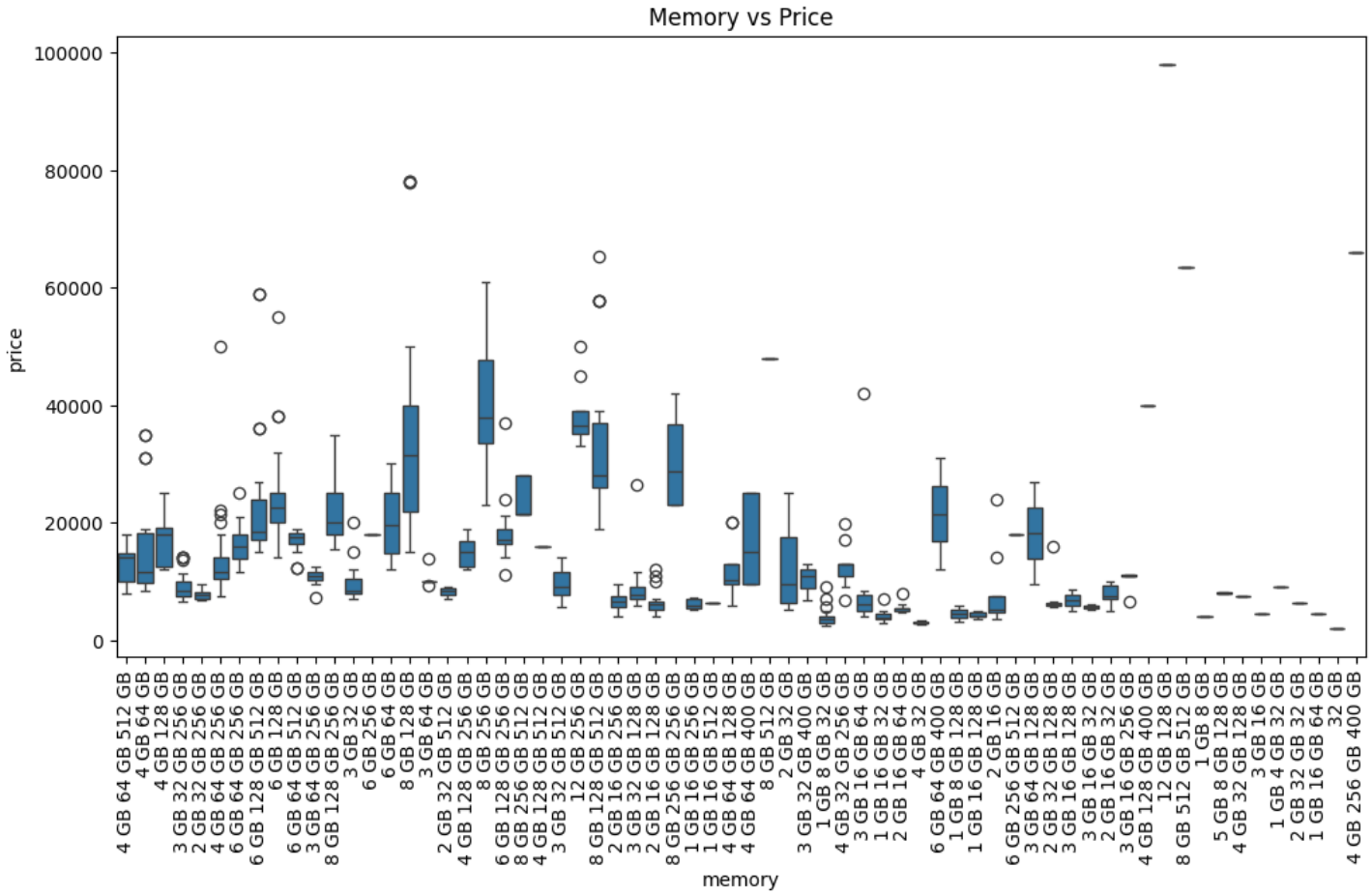


Overall, the strongest relationship is between price and rating, but none of the correlations are particularly strong, indicating that other factors might also be at play in influencing these variables.

In [24]:

```
# Boxplots to check the relationship between memory and price
plt.figure(figsize=(12, 6))
sns.boxplot(x='memory', y='price', data=df)
plt.title('Memory vs Price')
```

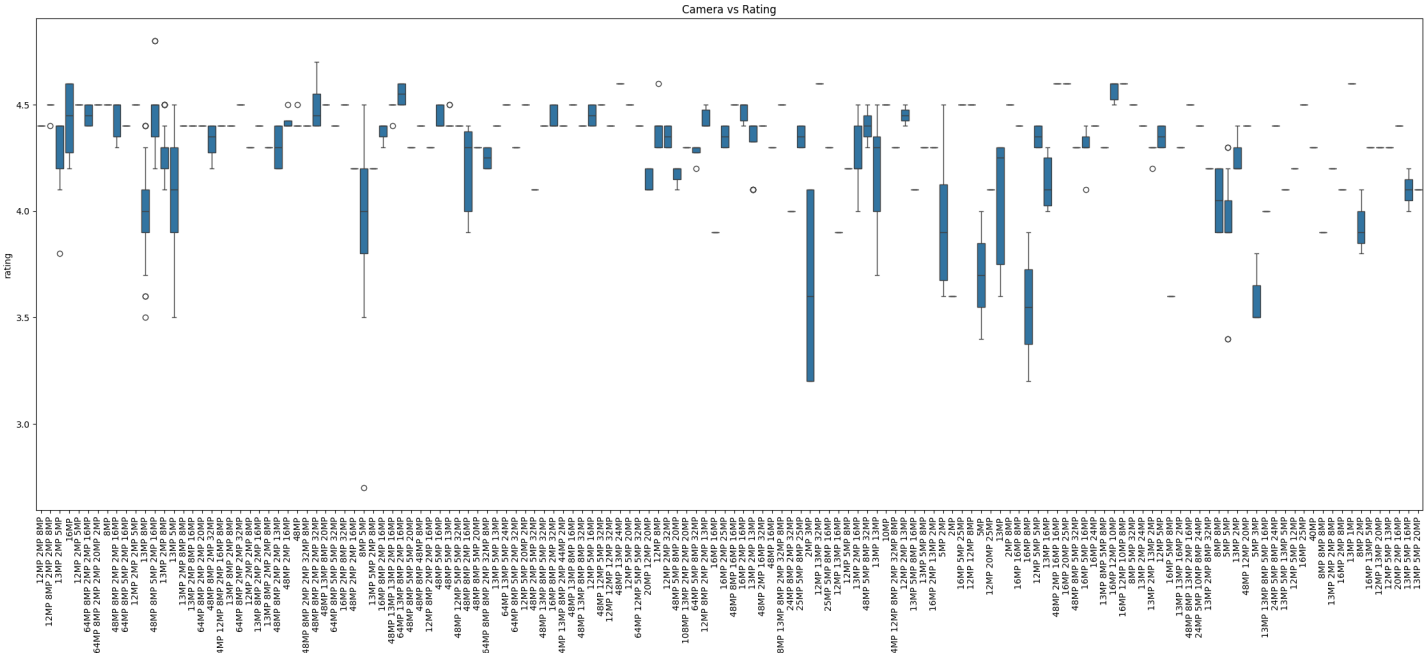
```
plt.xticks(rotation=90)
plt.show()
```



Price varies significantly with memory configurations, with higher memory generally associated with higher prices.Outliers are prevalent in many memory categories, indicating a wide range of pricing strategies within the same memory configuration.Some categories show skewed distributions, with prices leaning towards higher or lower extremes.

In [25]:

```
# Boxplots to check the relationship between categorical features and price/rating

plt.figure(figsize=(28, 10))
sns.boxplot(x='camera', y='rating', data=df)
plt.title('Camera vs Rating')
plt.xticks(rotation=90)
plt.show()
```



camera

Overall, while most cameras are rated positively, user experience can vary significantly for some models, and camera specifications alone don't determine user satisfaction.

In [ ]:

```
```

Bonus Task

ML

In [26]:

```
df.columns
```

Out[26]:

```
Index(['battery', 'camera', 'display', 'memory', 'name', 'price', 'processor',
       'rating', 'reviews', 'warranty'],
      dtype='object')
```

In [27]:

```
df.head()
```

Out[27]:

	battery	camera	display	memory	name	price	processor	rating	reviews	warranty
0	5000	12MP 2MP 8MP	15.8	4 GB 64 GB 512 GB	Redmi 8 (Ruby Red, 64 GB)	9999	Qualcomm Snapdragon 439 Processor	4.4	55078	Brand Warranty of 1 Year Available for Mobile ...
1	5000	12MP 8MP 2MP 2MP 8MP	16.56	4 GB 64 GB	Realme 5i (Aqua Blue, 64 GB)	10999	Qualcomm Snapdragon 665 2 GHz Processor	4.5	20062	
2	5000	12MP 8MP 2MP 2MP 8MP	16.56	4 GB 128 GB	Realme 5i (Aqua Blue, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20062	
3	5000	12MP 8MP 2MP 2MP 8MP	16.56	4 GB 128 GB	Realme 5i (Forest Green, 128 GB)	11999	Qualcomm Snapdragon 665 (2 GHz) Processor	4.5	20062	
4	4000	13MP 2MP 5MP	15.49	3 GB 32 GB 256 GB	Realme C2 (Diamond Blue, 32 GB)	7499	MediaTek P22 Octa Core 2.0 GHz Processor	4.4	10091	

In [28]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
```

In [29]:

```
df = df.fillna('0')
```

In [30]:

```
# Encoding categorical variables basis on hierarchy

for col in ['warranty',  'camera','memory','processor']:
```

```
    df[col] = LabelEncoder().fit_transform(df[col].astype(str))


#creating dummies for nominal categorical variables
df = pd.get_dummies(df, columns=['display', 'battery'])

# dropping unwanted columns
X = df.drop(columns=['rating', 'name'])
y = df['rating'].dropna()

#spliting into train_test split(80% train and 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)

#predicting the test data
y_pred = model.predict(X_test)

# Finding score
r2 = r2_score(y_test, y_pred)
r2
```

Out[30]:

```
0.7452556606114014
```

Here, our mode score is above avg i.e. 70+ means we can consider this model for predictive analysis

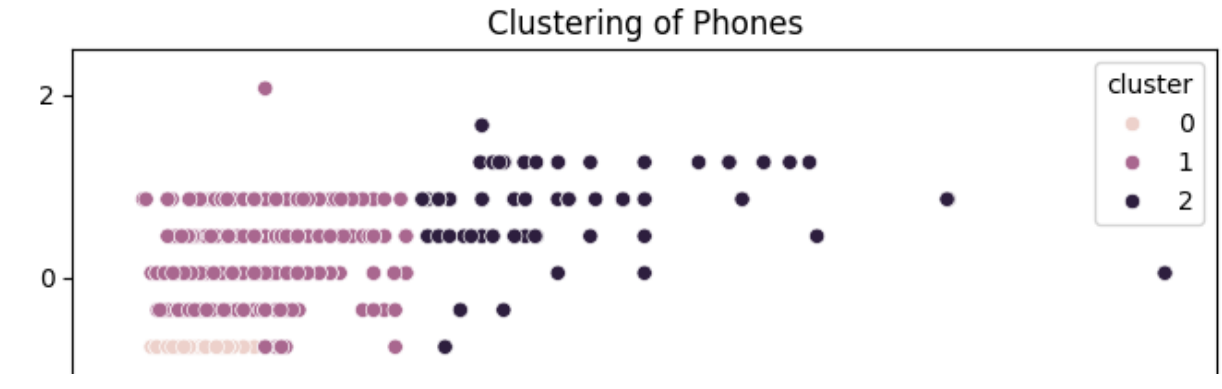
In [ ]:

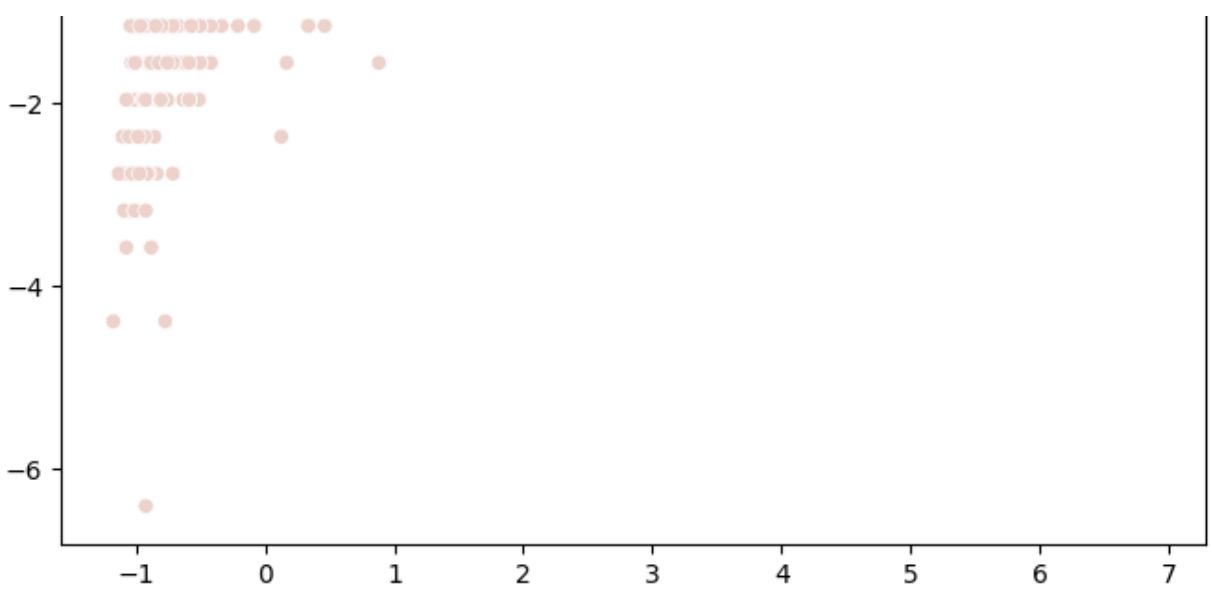
```
```

Clustering of phones

In [31]:

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Selecting features for clustering
clustering_data = df[['price',  'rating']]

# Scaling the data
scaler = StandardScaler()
clustering_data_scaled = scaler.fit_transform(clustering_data.dropna())

# Apply KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(clustering_data_scaled)

# Visualize clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x=clustering_data_scaled[:, 0], y = clustering_data_scaled[:, 1], hue=df
['cluster'])
plt.title('Clustering of Phones')
plt.show()
```





This analysis suggests that price and rating are moderately correlated, but there are exceptions, especially in the lower price ranges, where phones with similar ratings can belong to different clusters.

See you soon ;)

In [ ]: