my_assignment

August 30, 2024

[1]:

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

[3]:

```
data=pd.read_csv("Assignment - Junior Data Analyst.csv")
```

[4]:

```
data.head()
```

[4]:

```
            battery                                     camera  \
0  5000 mAh Battery              12MP + 2MP | 8MP Front Camera
1  5000 mAh Battery  12MP + 8MP + 2MP + 2MP | 8MP Front Camera
2  5000 mAh Battery  12MP + 8MP + 2MP + 2MP | 8MP Front Camera
3  5000 mAh Battery  12MP + 8MP + 2MP + 2MP | 8MP Front Camera
4  4000 mAh Battery              13MP + 2MP | 5MP Front Camera

                            display  \
0   15.8 cm (6.22 inch) HD+ Display
1  16.56 cm (6.52 inch) HD+ Display
2  16.56 cm (6.52 inch) HD+ Display
3  16.56 cm (6.52 inch) HD+ Display
4   15.49 cm (6.1 inch) HD+ Display

                                          memory  \
0  4 GB RAM | 64 GB ROM | Expandable Upto 512 GB
1                           4 GB RAM | 64 GB ROM
2                          4 GB RAM | 128 GB ROM
3                          4 GB RAM | 128 GB ROM
4  3 GB RAM | 32 GB ROM | Expandable Upto 256 GB

                               name  price  \
0         Redmi 8 (Ruby Red, 64 GB)   9999
1      Realme 5i (Aqua Blue, 64 GB)  10999
2     Realme 5i (Aqua Blue, 128 GB)  11999
3  Realme 5i (Forest Green, 128 GB)  11999
4   Realme C2 (Diamond Blue, 32 GB)   7499
```

```
                               processor  rating          reviews  \
0           Qualcomm Snapdragon 439 Processor     4.4  55,078 Reviews
1     Qualcomm Snapdragon 665 2 GHz Processor     4.5  20,062 Reviews
2   Qualcomm Snapdragon 665 (2 GHz) Processor     4.5  20,062 Reviews
3   Qualcomm Snapdragon 665 (2 GHz) Processor     4.5  20,062 Reviews
4    MediaTek P22 Octa Core 2.0 GHz Processor     4.4  10,091 Reviews

                                          warranty
0  Brand Warranty of 1 Year Available for Mobile …
1                                   Sunrise Design
2                                   Sunrise Design
3                                   Sunrise Design
4          Dual Nano SIM slots and Memory Card Slot
```

[5]:

```
data.describe()
```

[5]:

```
                price      rating
count     984.000000  971.000000
mean    15429.848577    4.241195
std     12891.355967    0.300296
min       887.000000    2.700000
25%      7499.000000    4.100000
50%     11649.000000    4.300000
75%     17999.250000    4.400000
max    104999.000000    4.900000
```

[6]:

```
df = pd.DataFrame(data)
```

[7]:

```
def  categorize_premium(price,rating):
    if price > 50000 and rating>=4.5:
        return 'High Premium'
    elif price < 50000 and rating>=4.5:
        return 'mid Premium'
    if price < 50000 and rating<4.5:
        return 'low Premium'
```

[10]:

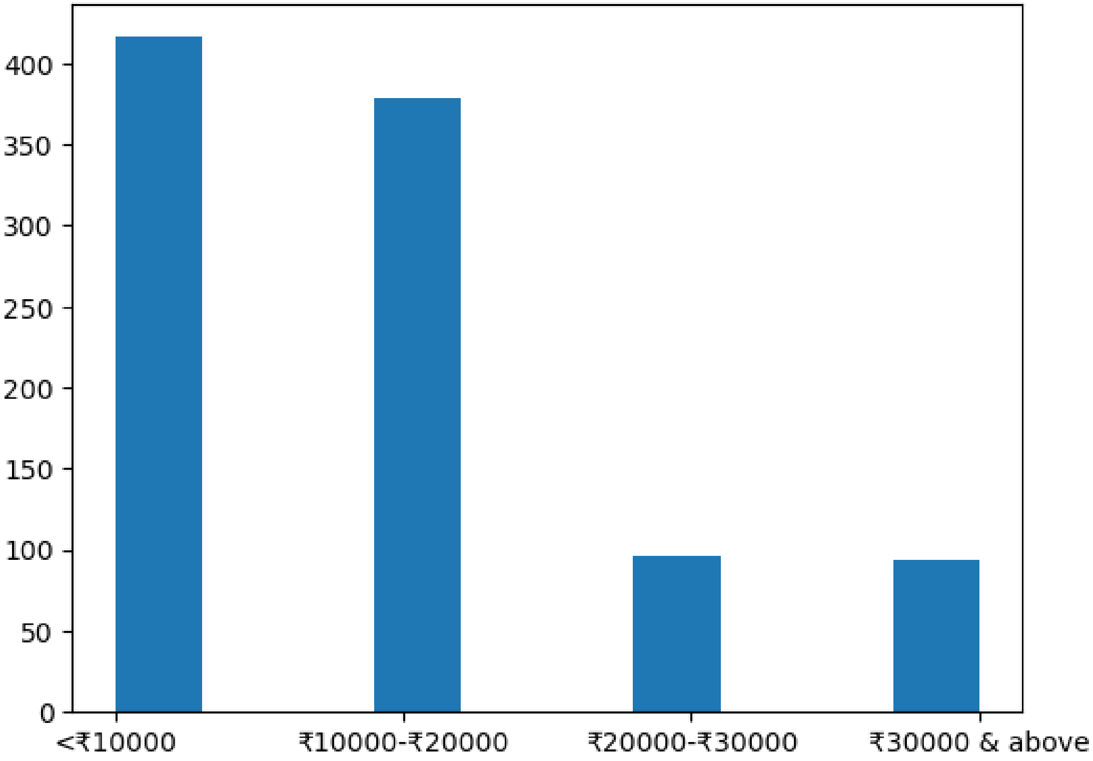
```
#creating calculated columns : Price_bins,Brand,cluster_phone,battery_capacity
df['Price_bins'] = df['price'].apply(Price_category)
df['Brand']=df['name'].apply(lambda x:x.split()[0])
df['cluster_phone']=df.apply(lambda row: categorize_premium(row['price'],␣
 ↪row['rating']), axis=1)
df['battery_capacity']=df['battery'].apply(lambda x:x.split()[0])
```

[9]:

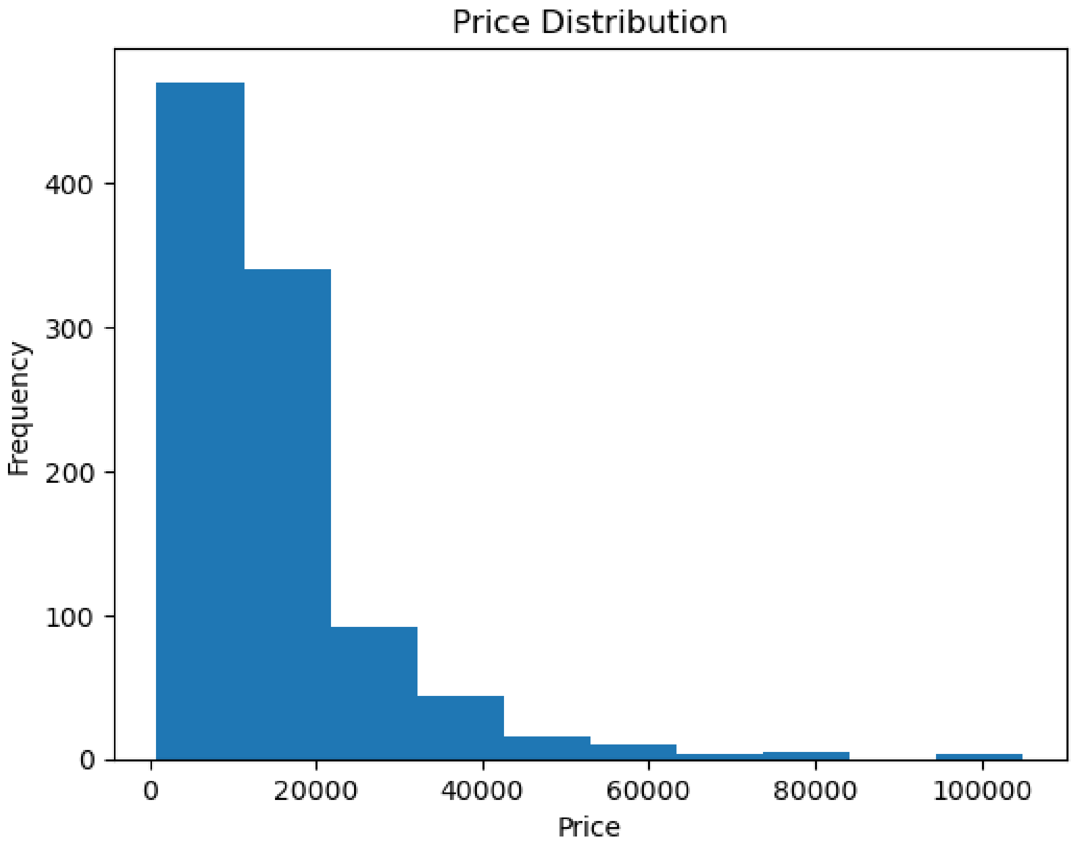
```
#univariant Analysis
def Price_category(price):
    if price < 10000:
        return '< 10000'
    elif 10000 <= price < 20000:
        return ' 10000- 20000'
    elif 20000 <= price < 30000:
        return ' 20000- 30000'
    else:
        return ' 30000 & above'
```

[11]:
```
x=df['Price_bins']
plt.hist(x)
```

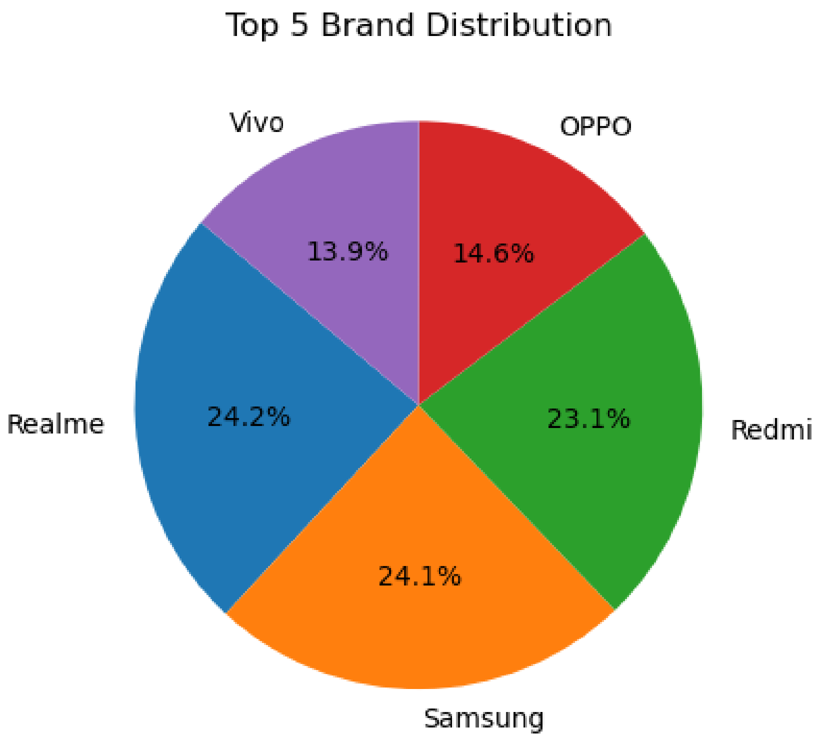
[11]:
```
(array([416.,   0.,   0., 378.,   0.,   0.,  96.,   0.,   0.,  94.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 <BarContainer object of 10 artists>)
```



[12]:
```
#Histogram: To visualize the distribution of phone prices
df['price'].plot(kind='hist', bins=10, title='Price Distribution')
plt.xlabel('Price')
plt.show()
```

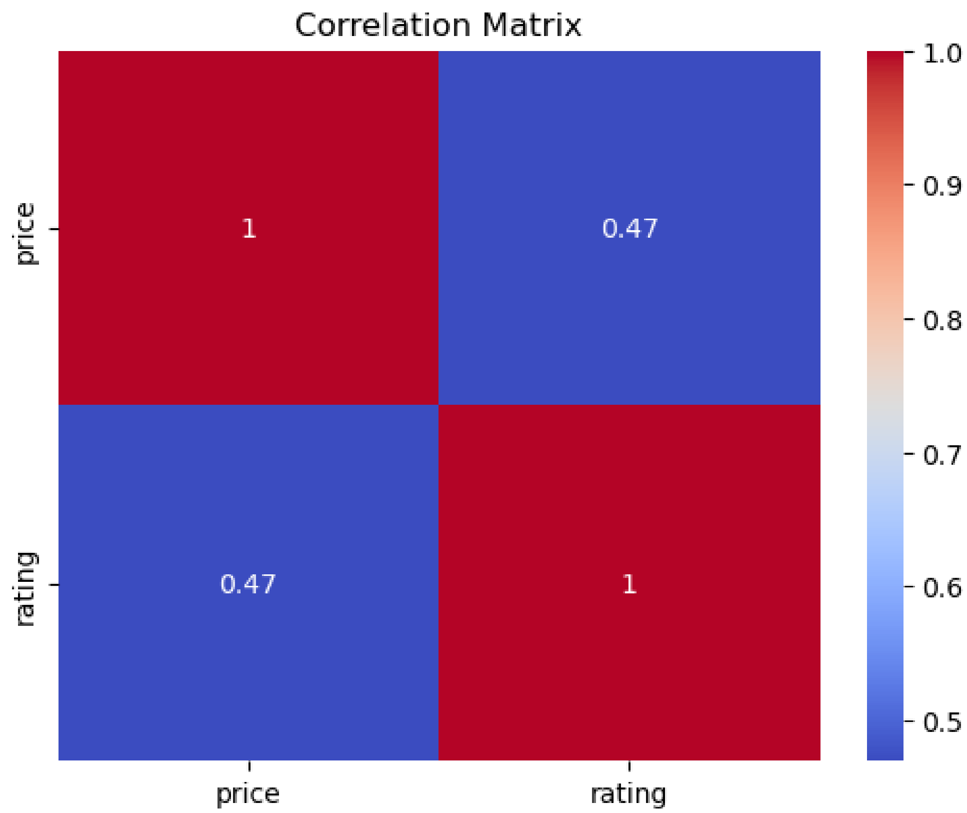


[13]:
```
top_5_brands = df['Brand'].value_counts().nlargest(5)
top_5_brands.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Top 5 Brand Distribution')
plt.ylabel('')
plt.show()
```



[14]:

```
#bivariant Analysis
correlation_matrix = df[['price', 'rating']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



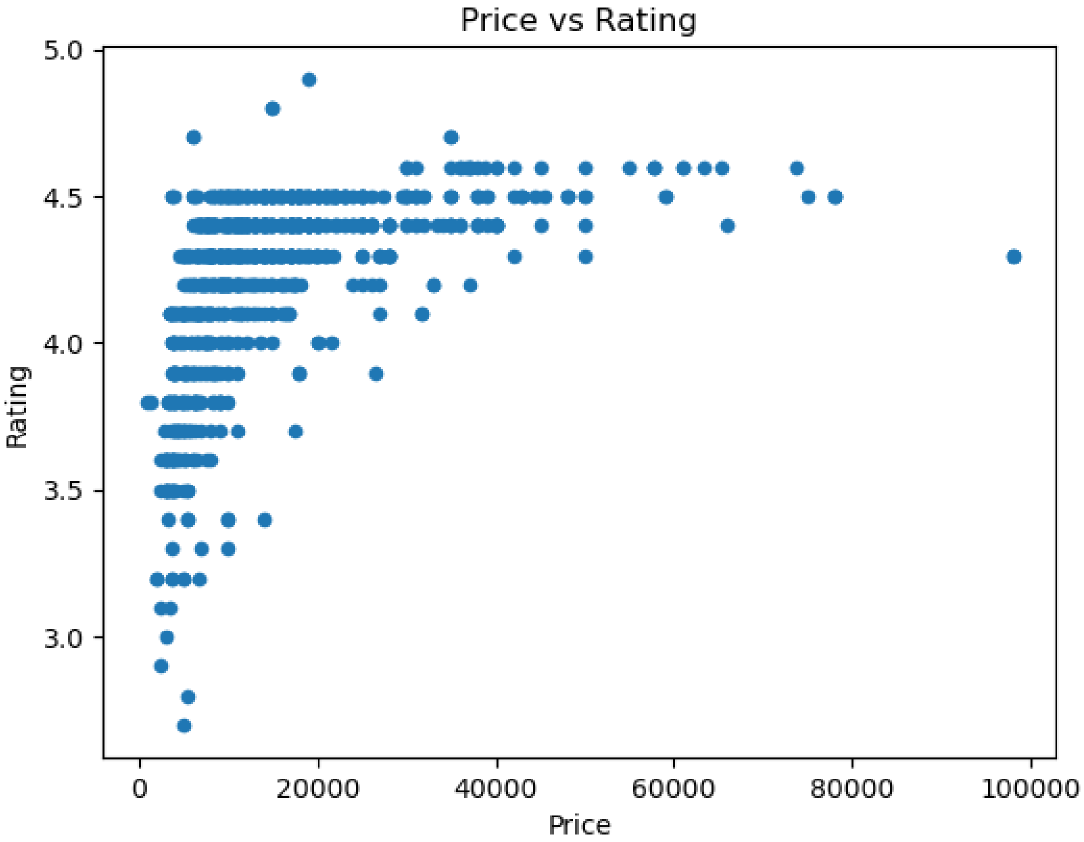
[15]:

```
correlation = df[['price', 'rating']].corr()
print(correlation)
```

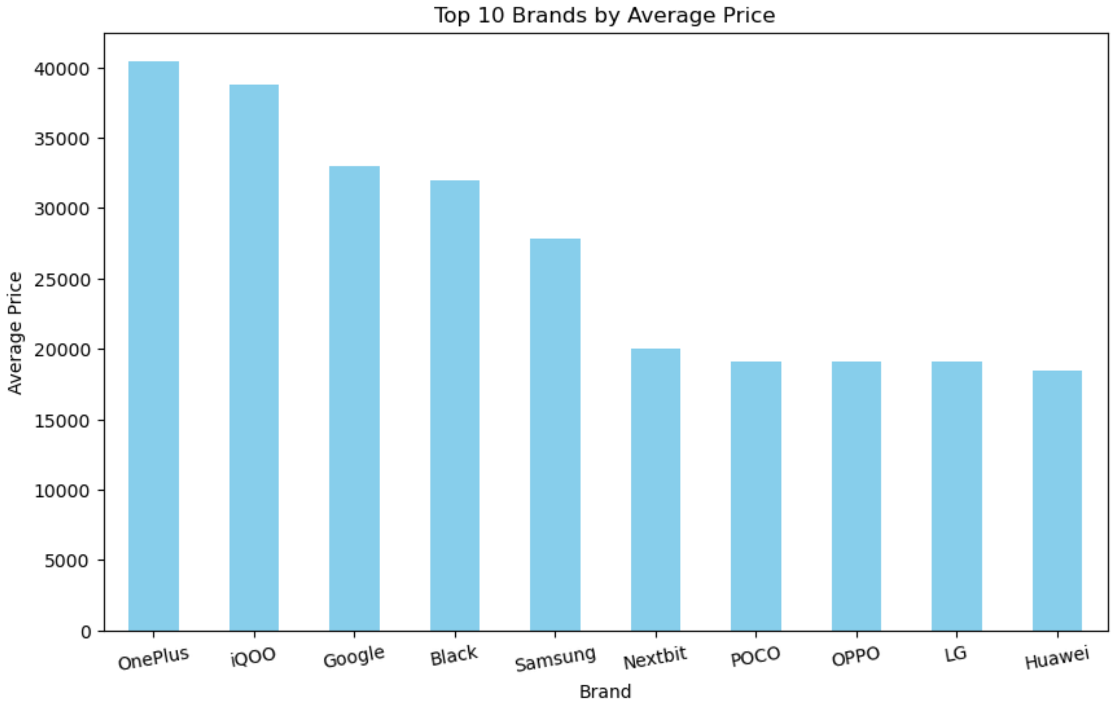
```
           price    rating
price   1.000000  0.469329
rating  0.469329  1.000000
```

[16]:

```
#Scatter Plot: To visualize the correlation between price and rating
df.plot(kind='scatter', x='price', y='rating', title='Price vs Rating')
plt.xlabel('Price')
plt.ylabel('Rating')
plt.show()
```



[19]:
```
avg_price_per_brand=df.groupby('Brand')['price'].mean()
top_10_brands_by_price = avg_price_per_brand.nlargest(10)
plt.figure(figsize=(10, 6))
top_10_brands_by_price.plot(kind='bar', color='skyblue')
plt.title('Top 10 Brands by Average Price')
plt.xlabel('Brand')
plt.ylabel('Average Price')
plt.xticks(rotation=10)
plt.show()
```


Top 10 Brands by Average Price
Average Price
40000
35000
30000
25000
20000
15000
10000
5000
0
OnePlus
iQOO
Google
Black
Samsung
Nextbit
POCO
OPPO
LG
Huawei
Brand

[ ]: