

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import mysql.connector

db = mysql.connector.connect(host = "your_host",
                             username = "your_username",
                             password = "your_password",
                             database = "your_database")

cur = db.cursor()
```

### 1. List all Mobile Brands available in the dataset

```
In [3]: query = """ select distinct Brands from mobilecompany.mobile; """

cur.execute(query)

data = cur.fetchall()

data
```

```
Out[3]: [('realme',),
('Redmi',),
('Samsung',),
('OnePlus',),
('Vivo',),
('Motorola',),
('Infinix',),
('Tecno',),
('Moto',),
('Oppo',),
('Xiaomi',),
('Itel',),
('Mi',),
('Apple',),
('Nokia',),
('JioPhone',)]
```

### 2. Find the average price of mobiles for each brand.

```
In [4]: query = """ select Brands, AVG(Price) as
              average_price FROM mobile
              GROUP BY Brands """

cur.execute(query)

query = cur.fetchall()

query
```

```
Out[4]: [('realme', Decimal('17818.5122')),
          ('Redmi', Decimal('15096.2222')),
          ('Samsung', Decimal('42763.9485')),
          ('OnePlus', Decimal('34868.5652')),
          ('Vivo', Decimal('19699.0175')),
          ('Motorola', Decimal('24808.5238')),
          ('Infinix', Decimal('12561.5000')),
          ('Tecno', Decimal('10299.0000')),
          ('Moto', Decimal('9999.0000')),
          ('Oppo', Decimal('16561.4737')),
          ('Xiaomi', Decimal('29158.0909')),
          ('Itel', Decimal('6082.1429')),
          ('Mi', Decimal('36707.3333')),
          ('Apple', Decimal('104682.1782')),
          ('Nokia', Decimal('12262.6364')),
          ('JioPhone', Decimal('4499.0000'))]
```

### 3.Retrieve the details of the most expensive mobile

```
In [5]: query = """ select * FROM mobile
              ORDER BY Price DESC
              Limit 1 """

cur.execute(query)

data = cur.fetchall()

data
```

```
Out[5]: [('Apple iPhone 14 Pro Max 1 TB, Silver',
          'Apple',
          189900,
          'Not Specified',
          'iOS',
          'Yes',
          'No',
          '12 MP',
          'Not Specified',
          'China',
          'Not Specified',
          'Between 1,50,000-2,00,000',
          'Not Specified',
          1)]
```

### 4. Find the total number of mobiles for each brands

```
In [6]: query = """ select Brands, COUNT(*) as
              total_mobile FROM mobile
              GROUP BY Brands ; """

cur.execute(query)

data = cur.fetchall()

df = pd.DataFrame(data, columns = ["Brands", "total_mobile"])

df
```

Out[6]:

	Brands	total_mobile
0	realme	82
1	Redmi	72
2	Samsung	97
3	OnePlus	23
4	Vivo	57
5	Motorola	21
6	Infinix	16
7	Tecno	4
8	Moto	2
9	Oppo	38
10	Xiaomi	44
11	Itel	7
12	Mi	12
13	Apple	101
14	Nokia	22
15	JioPhone	2

## 5. Get the brand with the highest average RAM

```
In [7]: query = """ select Brands, AVG(RAM_Storage) as
              average_RAM FROM mobile
              GROUP BY Brands
              ORDER BY average_RAM DESC
              LIMIT 1 ; """

cur.execute(query)

data = cur.fetchall()

data
```

Out[7]: [('OnePlus', 8.956521739130435)]

**6. List all models with battery capacity greater than 45000 mAh.**

```
In [8]: query = """ select * FROM mobile
          WHERE Battery_Capacity > 4000 """

cur.execute(query)

data = cur.fetchall()

data
```

```
Out[8]: [('Realme 9 Pro 5G 128 GB, 8 GB RAM, Sunrise Blue, Mobile Phone',
          'realme',
          20999,
          '128 GB',
          'Android',
          'Not Specified',
          'No',
          '16 MP',
          '8 GB',
          'Not Specified',
          '5000 mAh',
          'Between 20,000-30,000',
          'Between 4001 mAh-5000 mAh',
          1),
          ('Realme 9 Pro 5G 128 GB, 6 GB RAM, Sunrise Blue, Mobile Phone',
          'realme',
          18999,
          '128 GB',
          'Android',
          'Not Specified',
          'No',
          '16 MP',
          '8 GB',
          'Not Specified',
          '5000 mAh',
          'Between 20,000-30,000',
          'Between 4001 mAh-5000 mAh',
          1)]
```

**7. Find the Brand with the cheapest average price**

```
In [9]: query = """ select Brands, AVG(Price) as
          average_price FROM mobile
          GROUP BY Brands
          ORDER BY average_price ASC
          LIMIT 1 ; """

cur.execute(query)

data = cur.fetchall()

data
```

```
Out[9]: [('JioPhone', Decimal('4499.0000'))]
```

**8. Get the models that have atleast 128GB of storage and 6GB of RAM**

```
In [10]: query = """ select * FROM mobile
          WHERE Internal_Storage >= 128 AND RAM_Storage >= 6; """

cur.execute(query)

data = cur.fetchall()

data
```

```
Out[10]: [('Realme 9 Pro 5G 128 GB, 8 GB RAM, Sunrise Blue, Mobile Phone',
          'realme',
          20999,
          '128 GB',
          'Android',
          'Not Specified',
          'No',
          '16 MP',
          '8 GB',
          'Not Specified',
          '5000 mAh',
          'Between 20,000-30,000',
          'Between 4001 mAh-5000 mAh',
          1),
          ('Realme 9 Pro 5G 128 GB, 6 GB RAM, Sunrise Blue, Mobile Phone',
          'realme',
          18999,
          '128 GB',
          'Android',
          'Not Specified',
          'No',
          '16 MP',
          '8 GB',
          'Not Specified',
          '5000 mAh',
          'Between 20,000-30,000',
          'Between 4001 mAh-5000 mAh',
          1)]
```

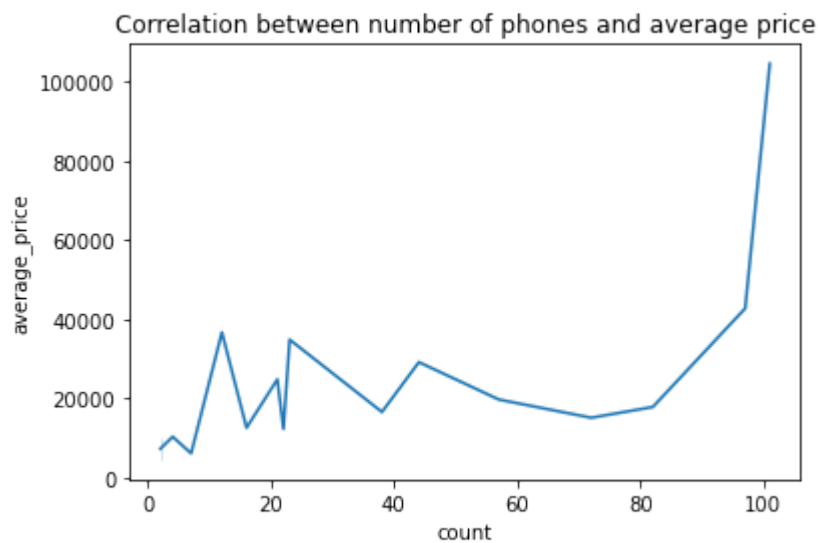
**9.How Does the number of mobile phones by brand correlate with their average price?**

```
In [11]: query = """ select Brands, AVG(Price) AS
                    average_price, COUNT(*) AS
                    count FROM
                    mobile
                    GROUP BY Brands"""

cur.execute(query)

data = cur.fetchall()

df = pd.DataFrame(data, columns = ['Brands', 'average_price', 'count'])
sns.lineplot(x = 'count', y = 'average_price', data = df)
plt.title('Correlation between number of phones and average price')
plt.show()
```



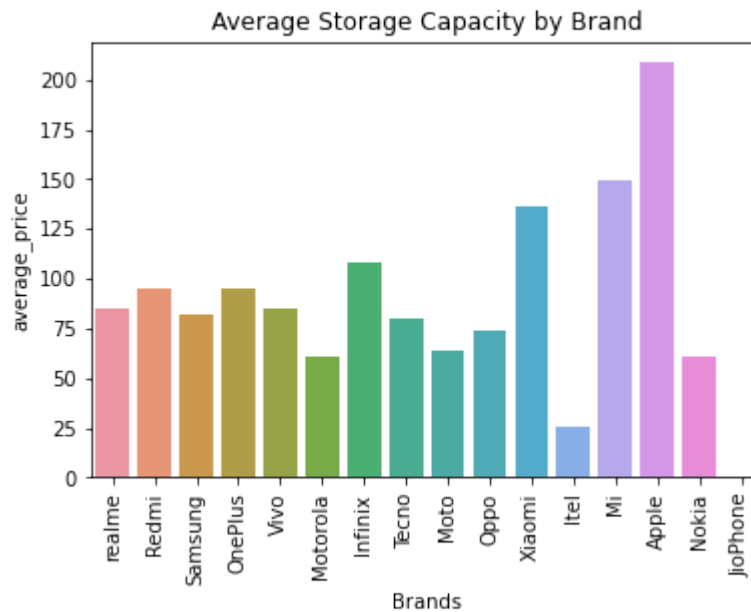
**10. How does the number of mobile phones by Brand Correlate with their Average Price?**

```
In [12]: query = """ select Brands, AVG(Internal_Storage) AS
                average_storage FROM mobile
                GROUP BY Brands"""

cur.execute(query)

data = cur.fetchall()

df = pd.DataFrame(data, columns = ['Brands', 'average_price'])
sns.barplot(x = 'Brands', y = 'average_price', data = df)
plt.title('Average Storage Capacity by Brand')
plt.xticks(rotation = 90)
plt.show()
```



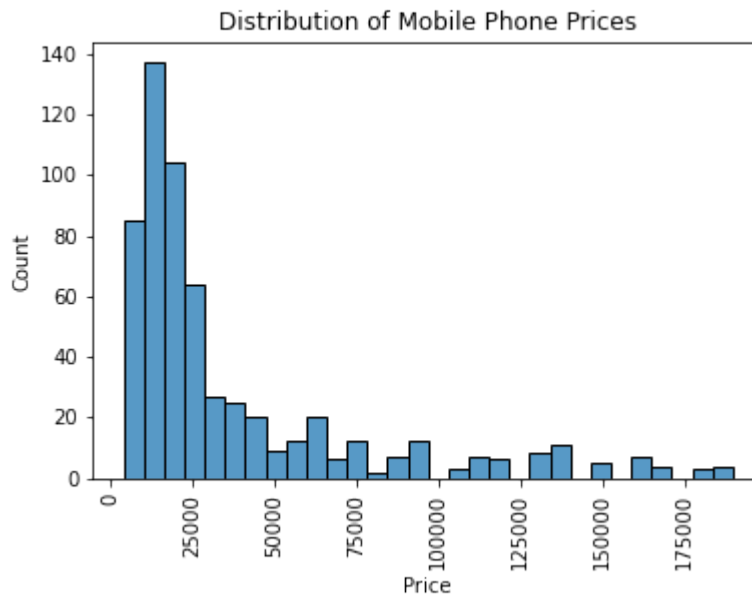
**11.What is the Distribution of Rating Accross all mobile phones?**

```
In [13]: query = """ select price FROM
            mobile"""

cur.execute(query)

data = cur.fetchall()

df = pd.DataFrame(data, columns = ['Price'])
sns.histplot(df['Price'], bins=30)
plt.title('Distribution of Mobile Phone Prices')
plt.xticks(rotation = 90)
plt.show()
```



In [ ]: