NAME: ADITYA SANAP

TOPIC: EXTRENSCLUB MINI PROJECT

EXPLORATORY DATA ANALYSIS (EDA) ON MOBILE PRICES 2023

```
#IMPORTING LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px
```

```
df = pd.read_csv(r'C:\ExternsClub Mini Project\Data for EDA.csv')
df
```

	Unnamed: 0	Phone Name	Rating ?/5	Number of Ratings	Back/Rare Camera	Front Camera	Processor	Price in INR	Date of Scraping	Company Name	Ram_size	Storage_size	Battery_size
0	0	POCO C50 (Royal Blue, 32 GB)	4.2	33561	8MP Dual Camera	5MP Front Camera	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	5649	2023-06-17	POCO	2.0	32.0	5000
1	1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77128	50MP + 2MP	8MP Front Camera	Mediatek Dimensity 700 Processor	11999	2023-06-17	POCO	4.0	64.0	5000
2	2	POCO C51 (Royal Blue, 64 GB)	4.3	15175	8MP Dual Rear Camera	5MP Front Camera	Helio G36 Processor	6999	2023-06-17	POCO	4.0	64.0	5000
3	3	POCO C55 (Cool Blue, 64 GB)	4.2	22621	50MP Dual Rear Camera	5MP Front Camera	Mediatek Helio G85 Processor	7749	2023-06-17	POCO	4.0	64.0	5000
4	4	POCO C51 (Power Black, 64 GB)	4.3	15175	8MP Dual Rear Camera	5MP Front Camera	Helio G36 Processor	6999	2023-06-17	POCO	4.0	64.0	5000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1285	1831	Infinix Note 7 (Forest Green, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1286	1832	Infinix Note 7 (Bolivia Blue, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1287	1833	Infinix Note 7 (Aether Black, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1288	1834	Infinix Zero 8i (Silver Diamond, 128 GB)	4.2	7117	48MP + 8MP + 2MP + AI Lens Camera	16MP + 8MP Dual Front Camera	MediaTek Helio G90T Processor	18999	2023-06-17	Infinix	8.0	128.0	4500
1289	1835	Infinix S5 (Quetzal Cyan, 64 GB)	4.3	15701	16MP + 5MP + 2MP + Low Light Sensor	32MP Front Camera	Helio P22 (MTK6762) Processor	10999	2023-06-17	Infinix	4.0	64.0	4000

1290 rows × 13 columns

CLEANING THE DATASET FOR FURTHER ANALYSIS

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

```
Index(['Phone Name', 'Rating ?/5', 'Number of Ratings', 'Back/Rare Camera',
       'Front Camera', 'Processor', 'Price in INR', 'Date of Scraping',
       'Company Name', 'Ram_size', 'Storage_size', 'Battery_size'],
      dtype='object')
```

```
df.isnull().sum()
```

```
Phone Name           0
Rating ?/5           0
Number of Ratings    0
Back/Rare Camera     0
Front Camera         0
Processor            0
Price in INR         0
Date of Scraping     0
Company Name         0
Ram_size             0
Storage_size         1
Battery_size         0
dtype: int64
```

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

```
df.isnull().sum().sum()
```

```
0
```

```
df.duplicated().sum()
```

```
39
```

```
df = df.drop_duplicates().reset_index()
df
```

HERE WE CAN SEE THAT ‘Storage_size’ COLUMNS HAS A NULL VALUE.

SO WE NEED TO PLACE SOME VALUE TO FILL THE SPACE SO THERE IS NO ERROR WHILE DOING ANALYSIS.

HERE WE HAVE PLACED “NaN” AT EMPTY PLACE.

BUT THERE ARE STILL SOME DUPLICATE VALUES.

	index	Phone Name	Rating ?/5	Number of Ratings	Back/Rare Camera	Front Camera	Processor	Price in INR	Date of Scraping	Company Name	Ram_size	Storage_size	Battery_size
0	0	POCO C50 (Royal Blue, 32 GB)	4.2	33561	8MP Dual Camera	5MP Front Camera	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	5649	2023-06-17	POCO	2.0	32.0	5000
1	1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77128	50MP + 2MP	8MP Front Camera	Mediatek Dimensity 700 Processor	11999	2023-06-17	POCO	4.0	64.0	5000
2	2	POCO C51 (Royal Blue, 64 GB)	4.3	15175	8MP Dual Rear Camera	5MP Front Camera	Helio G36 Processor	6999	2023-06-17	POCO	4.0	64.0	5000
3	3	POCO C55 (Cool Blue, 64 GB)	4.2	22621	50MP Dual Rear Camera	5MP Front Camera	Mediatek Helio G85 Processor	7749	2023-06-17	POCO	4.0	64.0	5000
4	4	POCO C51 (Power Black, 64 GB)	4.3	15175	8MP Dual Rear Camera	5MP Front Camera	Helio G36 Processor	6999	2023-06-17	POCO	4.0	64.0	5000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1246	1285	Infinix Note 7 (Forest Green, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1247	1286	Infinix Note 7 (Bolivia Blue, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1248	1287	Infinix Note 7 (Aether Black, 64 GB)	4.3	25582	48MP + 2MP + 2MP + AI Lens Camera	16MP Front Camera	MediaTek Helio G70 Processor	14999	2023-06-17	Infinix	4.0	64.0	5000
1249	1288	Infinix Zero 8i (Silver Diamond, 128 GB)	4.2	7117	48MP + 8MP + 2MP + AI Lens Camera	16MP + 8MP Dual Front Camera	MediaTek Helio G90T Processor	18999	2023-06-17	Infinix	8.0	128.0	4500
1250	1289	Infinix S5 (Quetzal Cyan, 64 GB)	4.3	15701	16MP + 5MP + 2MP + Low Light Sensor	32MP Front Camera	Helio P22 (MTK6762) Processor	10999	2023-06-17	Infinix	4.0	64.0	4000

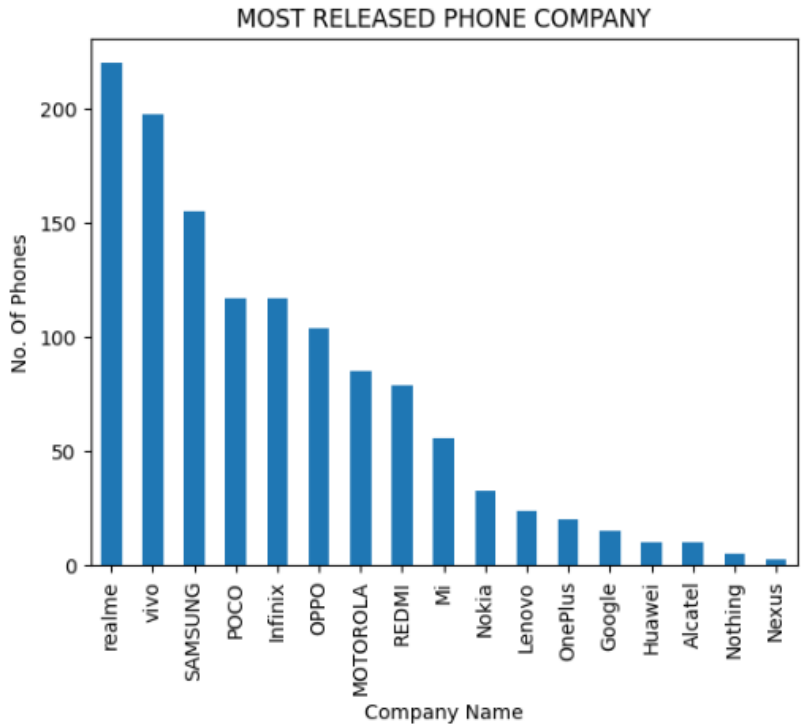
1251 rows × 13 columns

THE DATASET/ DATA IS NOW CLEARED AS IT DOESN'T HAVE ANY NULL VALUES OR DUPLICATE VALUES.

DATA IS READY FOR FURTHER ANALYSIS.

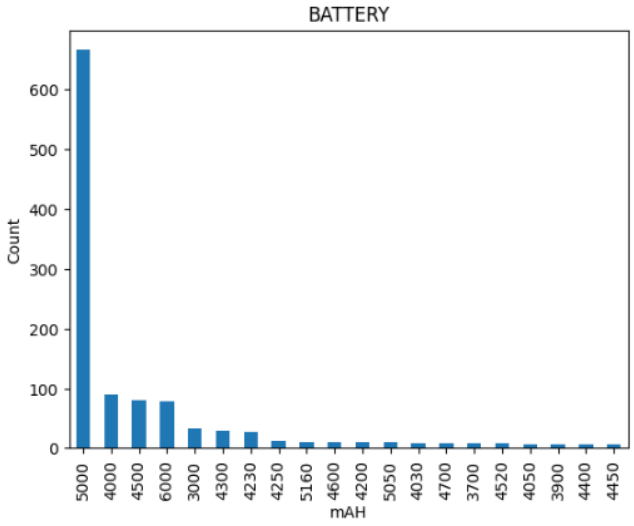
HERE ARE THE OBSERVATIONS/CONCLUSIONS:

```
df['Company Name'].value_counts().plot(kind = "bar", xlabel = "Company Name", ylabel = "No. Of Phones",
                                       title = "MOST RELEASED PHONE COMPANY")
plt.show()
```



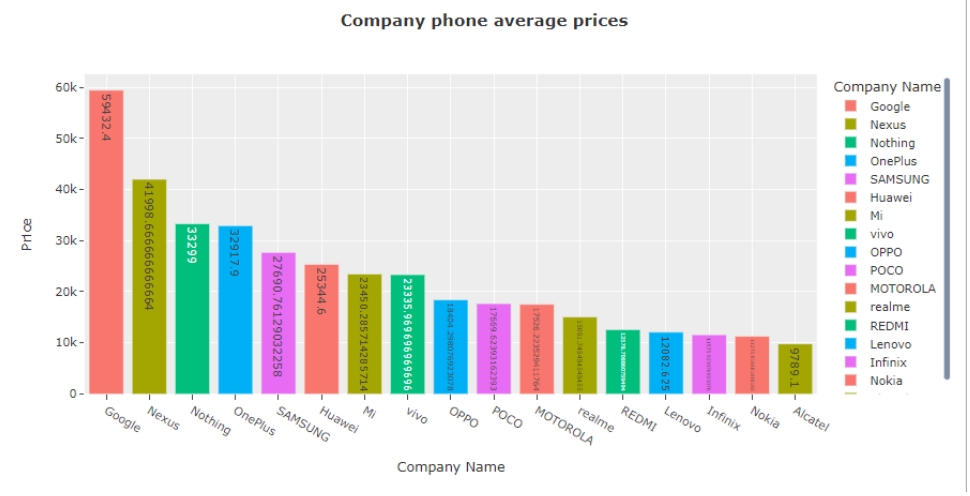
1. **REALME HAS RELEASED MOST NUMBER OF PHONES i.e 200+ FOLLOWING WITH VIVO & SAMSUNG.**
2. **NEXUS HAS RELEASED LESS 10 PHONES.**

```
df["Battery_size"].value_counts().head(20).plot(kind = "bar", xlabel = "mAH", ylabel = "Count", title = "BATTERY")
plt.show()
```



1. **MOST OF THE DEVICES/PHONES HAVE 5000 mAH BATTERY LIFE.**

```
fig=px.bar(df.groupby('Company Name',as_index=False)['Price in INR'].mean().sort_values(by='Price in INR',ascending=False),
           x='Company Name',y='Price in INR',color='Company Name',labels={'Company Name':'Company Name','Price in INR':'Price'},
           template='ggplot2',text='Price in INR',title='<b> Company phone average prices')
fig.show()
```

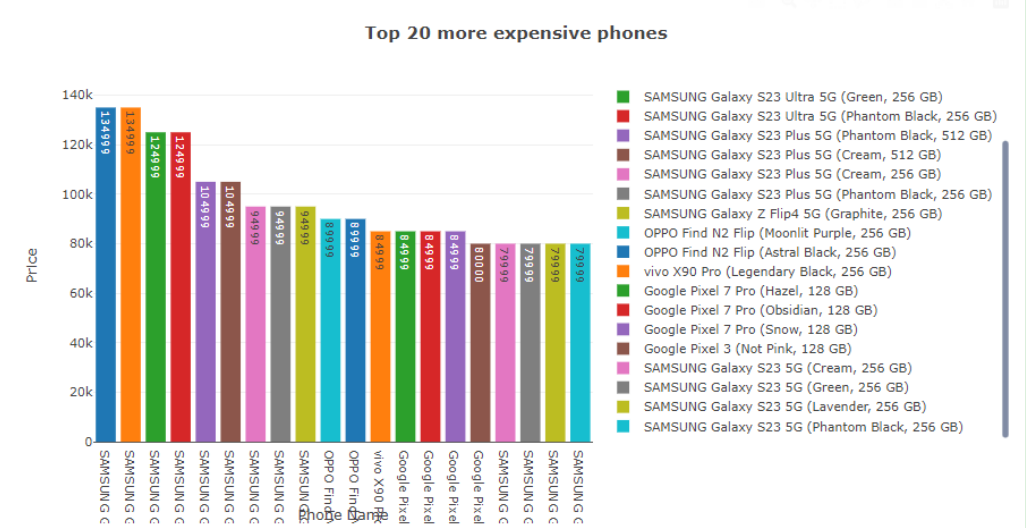


1. GOOGLE PHONES HAS MORE AVERAGE PRICE THAN OTHERS WITH ALMOST 60K INR.

2. NOTHING PHONES ARE LESS EXPANSIVE THEN NEXUS WHICH IS MORE THAN 40K.

3. ALCATEL PHONES ARE CHEAPEST FOLLOWED BY NOKIA

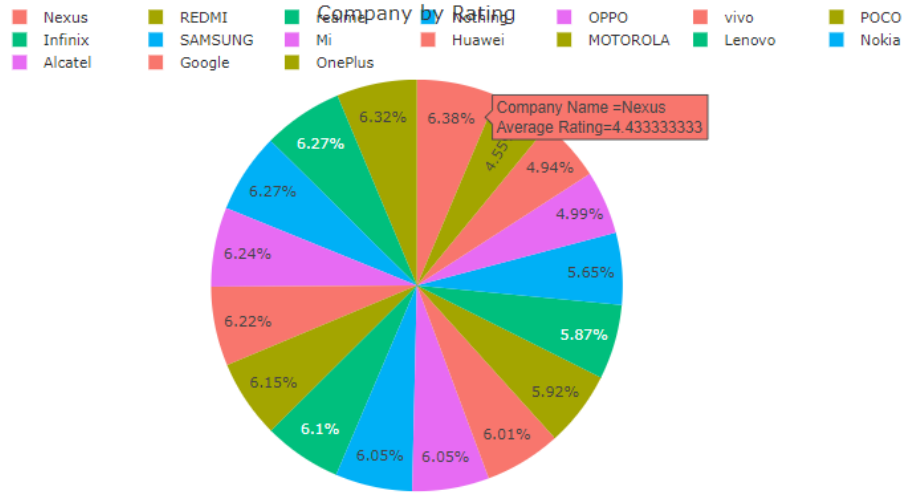
```
fig=px.bar(df.groupby('Phone Name',as_index=False)['Price in INR'].mean().sort_values(by='Price in INR',ascending=False).head(20)
           x='Phone Name',y='Price in INR',color='Phone Name',labels={'Phone Name':'Phone Name','Price in INR':'Price'},
           template='xgridoff',text='Price in INR',title='<b> Top 20 more expensive phones')
fig.show()
```



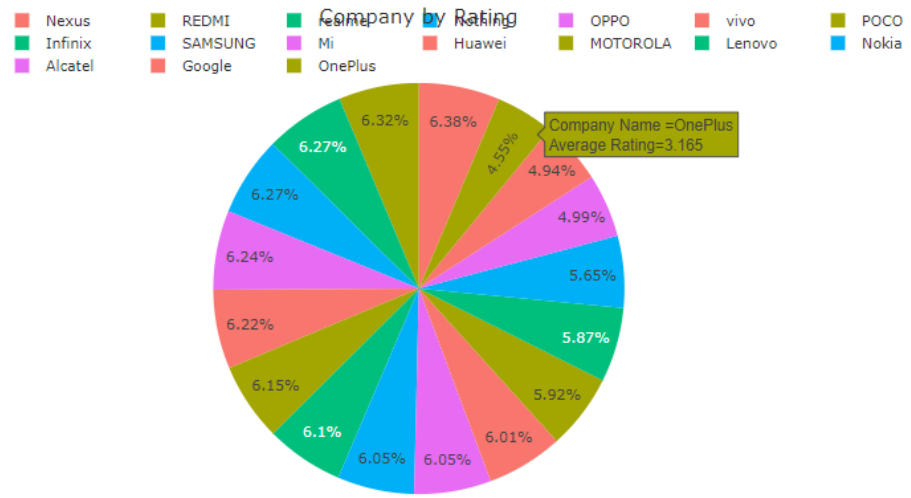
1. SAMSUNG PHONE ARE MORE EXPANSIVE LIKE GALAXY S23 ULTRA 5G

2. MOSTLY PHONES ARE SAMSUNG PHONES BUT SOME ARE LIKE OPPO, VIVO AND GOOGLE ARE ALSO INCLUDED INTO THIS LIST.

```
fig=px.pie(df.groupby('Company Name',as_index=False)['Rating ?/5'].mean().sort_values(by='Rating ?/5',ascending=False),
           names='Company Name',values='Rating ?/5',color='Company Name',
           labels={'Company Name':'Company Name ','Rating ?/5':'Average Rating'},template='ggplot2',
           title='<b>Company by Rating')
fig.update_layout(title_x=0.5,legend=dict(orientation='h',yanchor='bottom',
                                          y=1.0,xanchor='right',x=1))
```



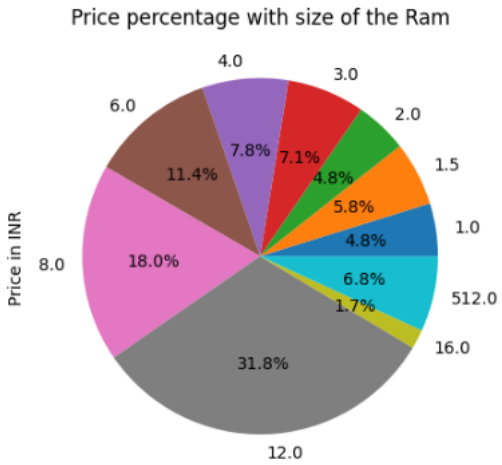

```
fig=px.pie(df.groupby('Company Name',as_index=False)['Rating ?/5'].mean().sort_values(by='Rating ?/5',ascending=False),
           names='Company Name',values='Rating ?/5',color='Company Name',
           labels={'Company Name':'Company Name ','Rating ?/5':'Average Rating'},template='ggplot2',
           title='<b>Company by Rating')
fig.update_layout(title_x=0.5,legend=dict(orientation='h',yanchor='bottom',
                                          y=1.0,xanchor='right',x=1))
```



1. **NEXUS COMPANY HAS THE MAXIMUM AVERAGE RATING OF 4.43333**
2. **ONEPLUS HAS THE LOWEST AVERAGE RATING OF 3.165**

```
df.groupby(['Ram_size'])['Price in INR'].mean().plot(kind='pie',autopct='%1.1f%%',title="Price percentage with size of the Ram")
```

```
<AxesSubplot: title={'center': 'Price percentage with size of the Ram'}, ylabel='Price in INR'>
```



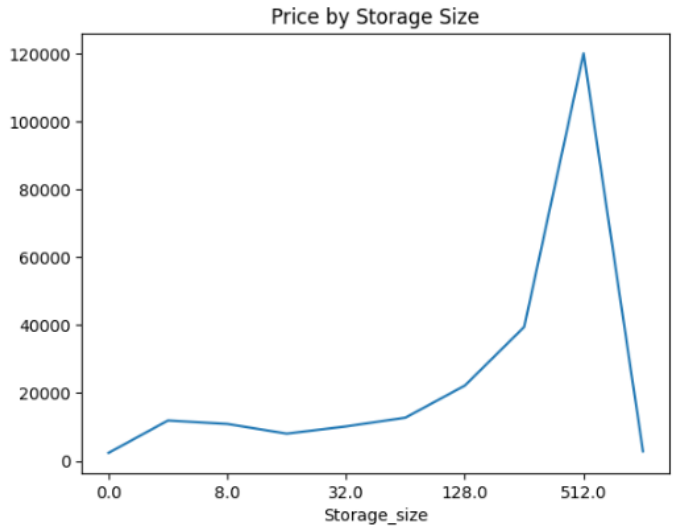
1. HERE IS THE 12.0 GB RAM IS MORE EXPANSIVE

2. AND 16 AND 512 LOOKS LIKE THE OUTLIERS IN THIS DATASET

3. AND THE CHEAPEST IS 1.0 GB RAM WHICH AS RAM SIZE INCREASES THE PRICES ALSO GOES UP.

```
df.groupby(['Storage_size'])['Price in INR'].mean().plot(kind='line', xlabel='Storage_size',title="Price by Storage Size")
```

```
<AxesSubplot: title={'center': 'Price by Storage Size'}, xlabel='Storage_size'>
```



1. **IT IS CLEARLY VISIBLE THAT THE UPWARD TREND MEANS THE PRICE INCREASES AS STORAGE PRICE INCREASES.**
2. **AS THERE IS NO STORAGE SIZE GREATER THEN 512GB PRICE DROPS.**