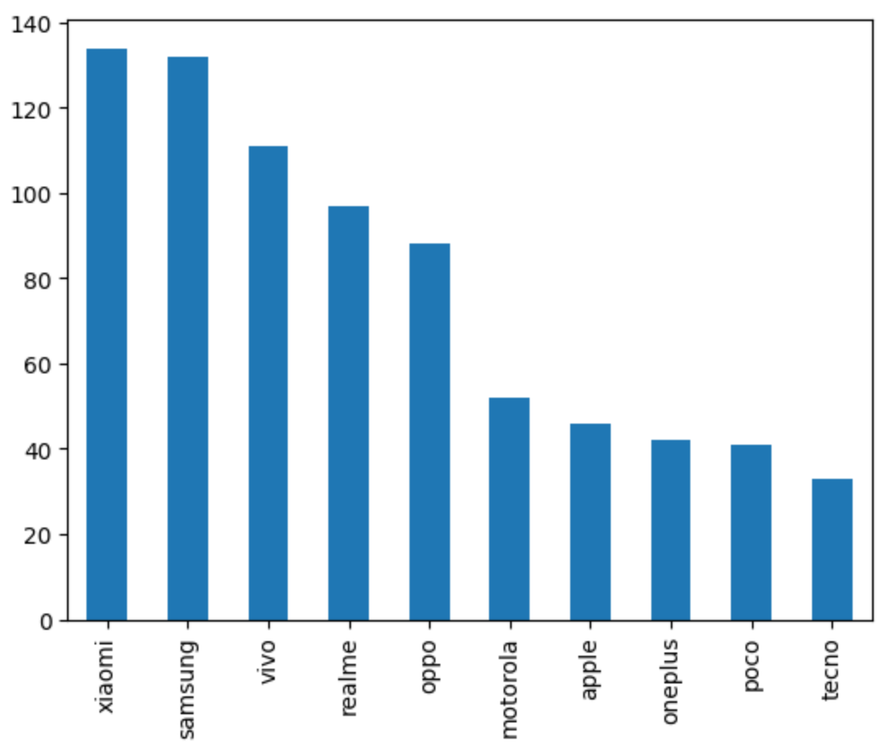
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)


df = pd.read_csv('/content/smartphone_cleaned_v5.csv')


df.head()
```

	brand_name	model	price	rating	has_5g	has_nfc	has_ir_blaster	processor_
0	oneplus	OnePlus 11 5G	54999	89.0	True	True	False	snap
1	oneplus	OnePlus Nord CE 2 Lite 5G	19989	81.0	True	False	False	snap
2	samsung	Samsung Galaxy A14 5G	16499	75.0	True	False	False	
3	motorola	Moto G62 5G	14999	81.0	True	False	False	snap
4	realme	Realme 10 Pro Plus	24999	82.0	True	False	False	din

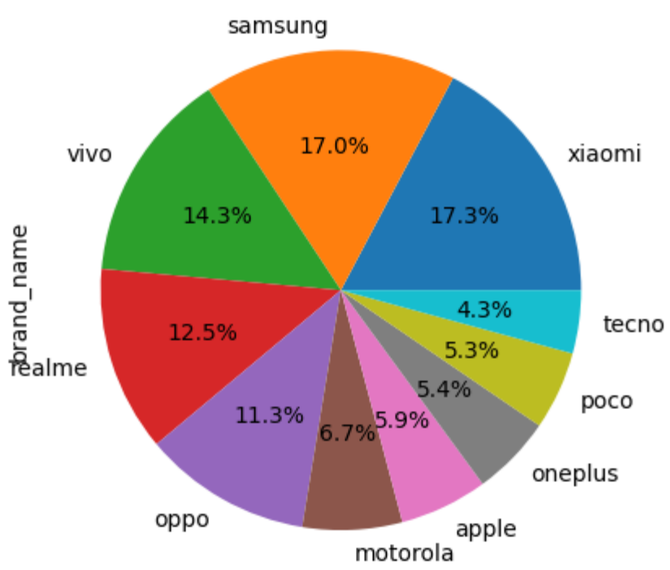
```
#Univariant Analysis
#brand


#top 10 brands with respect to models available in market


df['brand_name'].value_counts().head(10).plot(kind='bar')
```

```
<Axes: >
```




```
df['brand_name'].value_counts().head(10).plot(kind='pie',autopct='%0.1f%%')
```

```
<Axes: ylabel='brand_name'>
```




```
#model


df['model'].nunique()
```

```
980
```

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

```
brand_name                0
model                     0
price                     0
rating                  101
has_5g                    0
has_nfc                   0
has_ir_blaster            0
processor_brand          20
num_cores                 6
```

```
processor_speed               42
battery_capacity              11
fast_charging_available        0
fast_charging                211
ram_capacity                   0
internal_memory                0
screen_size                    0
refresh_rate                   0
resolution                     0
num_rear_cameras               0
num_front_cameras              4
os                            14
primary_camera_rear            0
primary_camera_front           5
extended_memory_available      0
extended_upto                480
dtype: int64
```

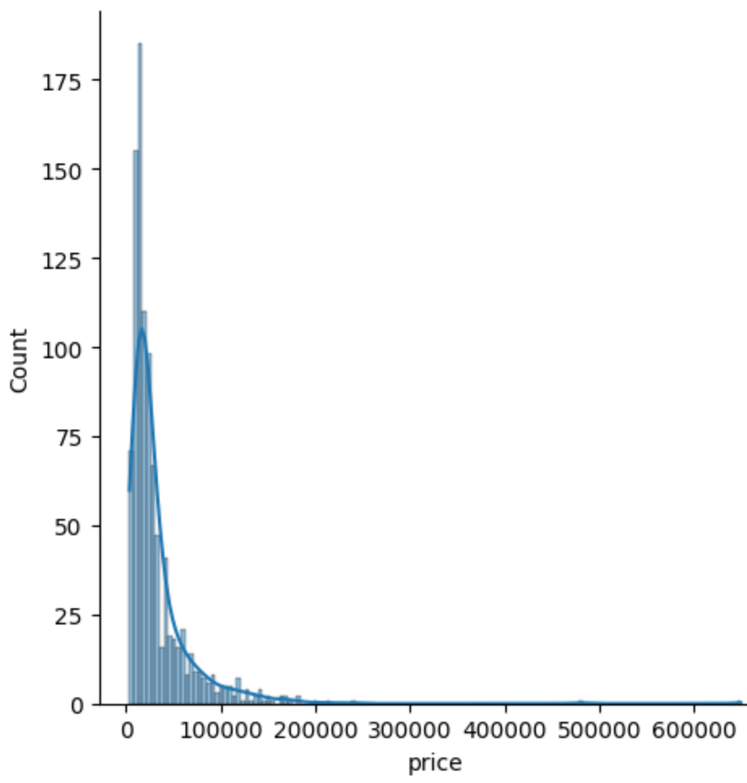
```
#price(Numeriacal Col)
```

```
df['price'].describe()
```

```
count       980.000000
mean      32520.504082
std       39531.812669
min        3499.000000
25%       12999.000000
50%       19994.500000
75%       35491.500000
max      650000.000000
Name: price, dtype: float64
```

```
#Checking Distribution of price data
sns.displot(kind='hist',data=df,x=df['price'],kde=True)

#conclusion:-More phones have less price and less phones have more price(Skewed Data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fad2579df90>
```

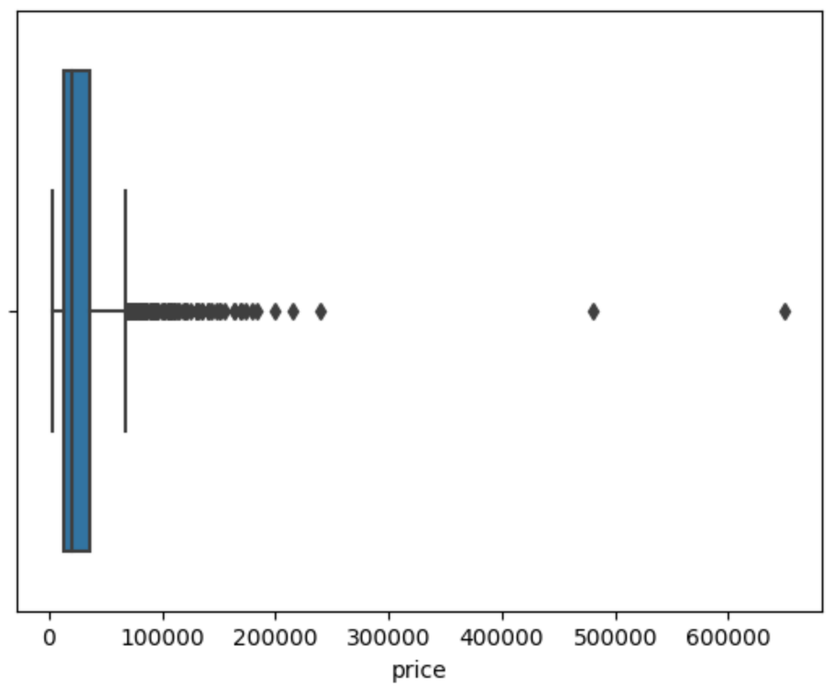



```
df['price'].skew()
```

```
6.591790999665567
```

```
sns.boxplot(x=df['price'])
```

```
<Axes: xlabel='price'>
```




```
df[df['price']>200000]
```

	brand_name	model	price	rating	has_5g	has_nfc	has_ir_blaster	processo
427	vertu	Vertu Signature Touch	650000	62.0	False	True	False	sn
		Huawei Mate 50						

#Conclusion
''' Price column is heavily skewed
there are some phones which can impact machine learning model prediction because this phones price is very high because they are gold plated and diamond integrated'''

887	xiaomi	K20 Pro	480000	88.0	False	True	False	sn

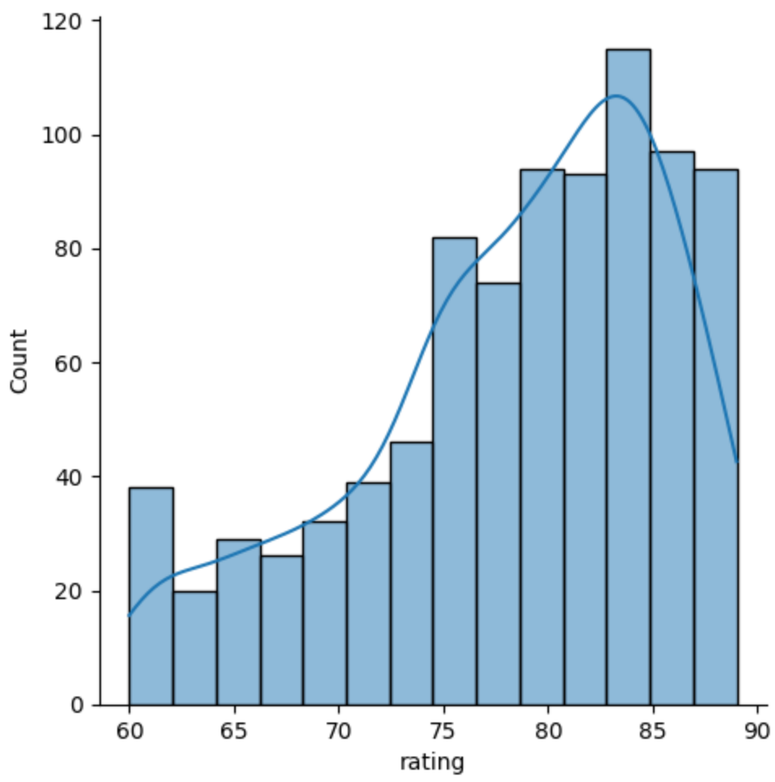
#Rating

```
df['rating'].describe()
```

```
count    879.000000
mean      78.258248
std        7.402854
min       60.000000
25%       74.000000
50%       80.000000
75%       84.000000
max       89.000000
Name: rating, dtype: float64
```

```
sns.displot(kind='hist',data=df,x='rating',kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fad1e1036a0>
```

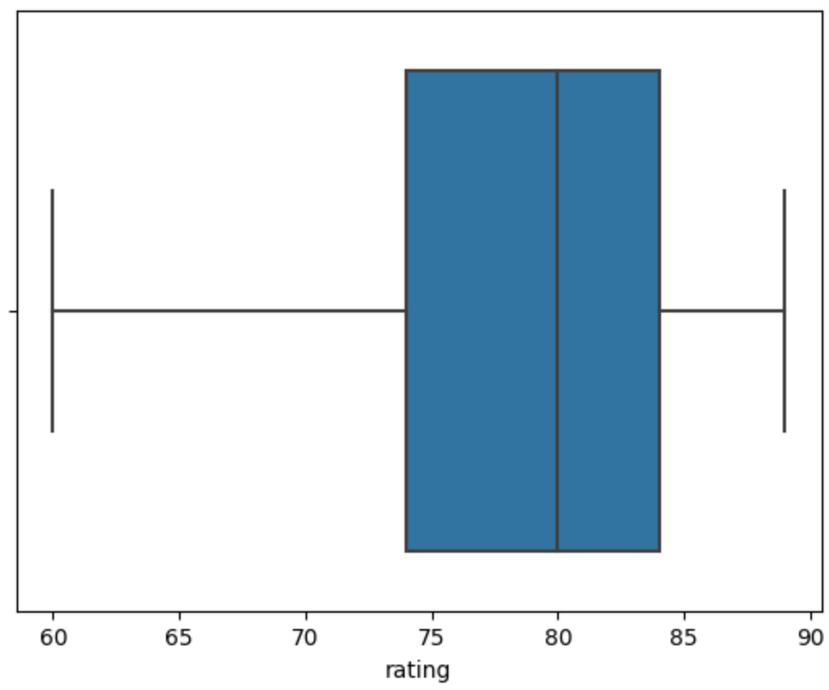



```
df['rating'].skew()
#Skewness is near to normal i.e (0)
```

```
-0.6989993034105535
```

```
sns.boxplot(x=df['rating'])
```

```
<Axes: xlabel='rating'>
```



#has_5g

```
df.head(1)
```

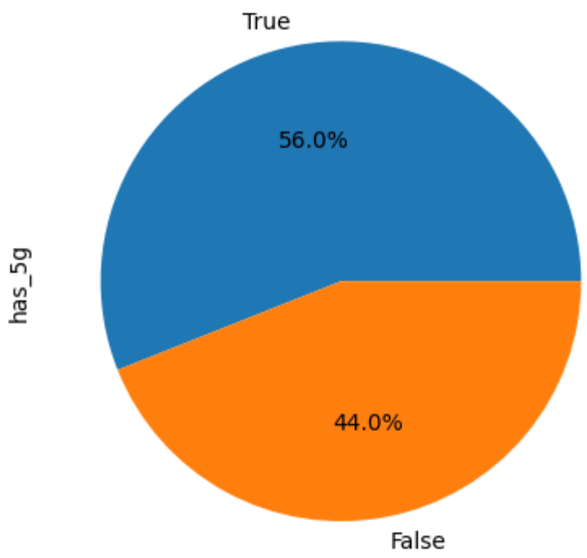
	brand_name	model	price	rating	has_5g	has_nfc	has_ir_blaster	processor_b
0	oneplus	OnePlus 11 5G	54999	89.0	True	True	False	snapd

```
df['has_5g'].value_counts()
```

```
True     549
False    431
Name: has_5g, dtype: int64
```

```
#how many percent of phones have 5g
df['has_5g'].value_counts().plot(kind='pie',autopct='%0.1f%%')
```

```
<Axes: ylabel='has_5g'>
```

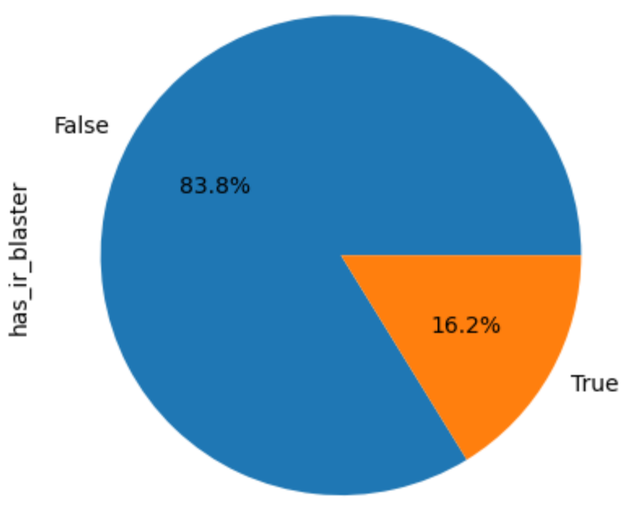



```
#has_ir_blaster


#how many percent of phones have ir_blaster
df['has_ir_blaster'].value_counts()
```

```
False    821
True     159
Name: has_ir_blaster, dtype: int64
```

```
df['has_ir_blaster'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:16.2 % of phones have ir_blaster
```

```
<Axes: ylabel='has_ir_blaster'>
```



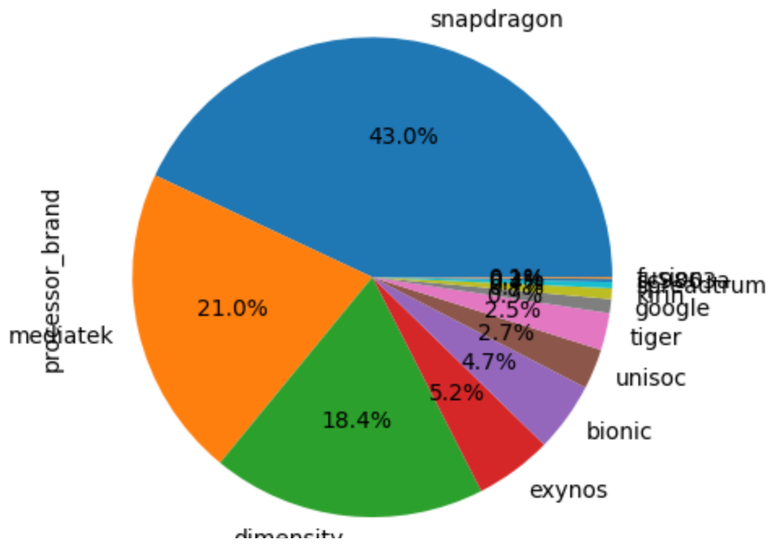

```
df[df['has_ir_blaster']==True]['brand_name'].value_counts()
#Conclusion=Only Chinese brand have ir_blaster in there phones
```

```
xiaomi     109
poco        30
iqoo         6
huawei       6
vivo         4
redmi        2
honor        1
samsung      1
Name: brand_name, dtype: int64


#processor_brand


df['processor_brand'].value_counts()
```

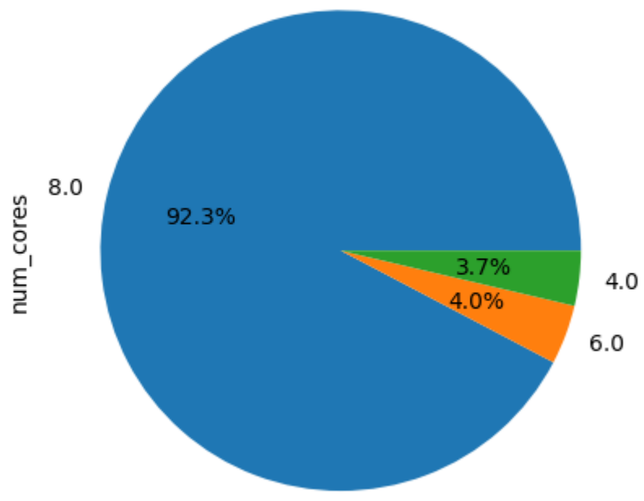
```
snapdragon    413
helio         201
dimensity     177
exynos         50
bionic         45
unisoc         26
tiger          24
google          9
kirin           7
spreadtrum      4
sc9863a         2
fusion          1
mediatek        1
Name: processor_brand, dtype: int64


#As helio and mediatek is same we will replace all helio values with mediatek
df['processor_brand']=df['processor_brand'].str.replace('helio','mediatek')


df['processor_brand'].value_counts()
```

```
snapdragon    413
mediatek      202
dimensity     177
exynos         50
bionic         45
unisoc         26
tiger          24
google          9
kirin           7
spreadtrum      4
sc9863a         2
fusion          1
Name: processor_brand, dtype: int64


df['processor_brand'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Snapdragon is most used processor in market following mediatek--dimensity---exynos
```

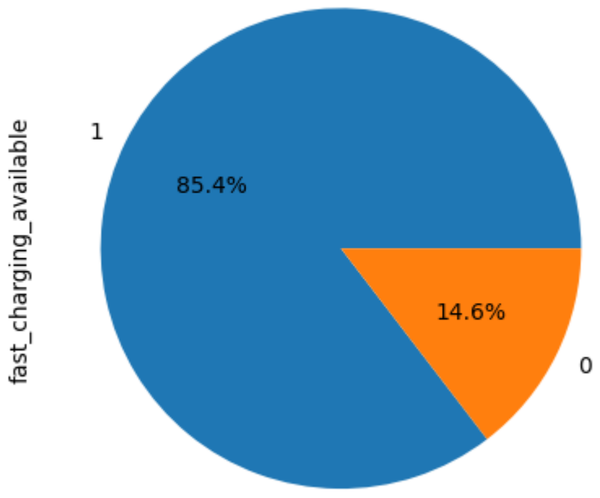
```
<Axes: ylabel='processor_brand'>
```



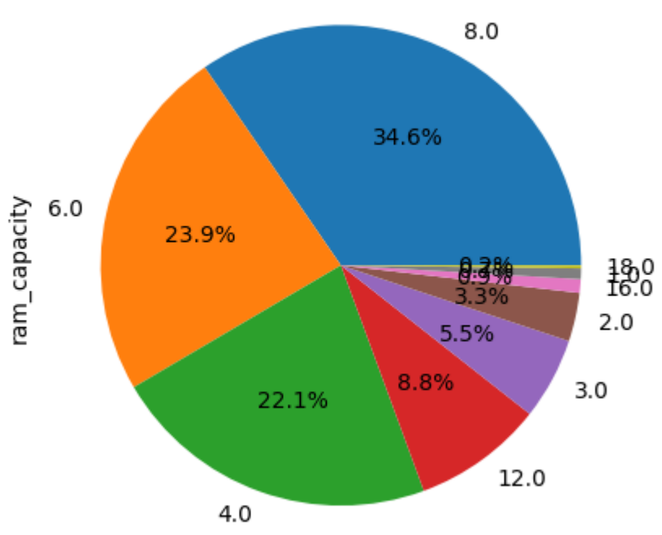

```
#num_cores


df['num_cores'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Most of the mobile phones companies are using octa-core processor(8.0) and 4% is using hexa_core and 3.7%is using quad_core

     <Axes: ylabel='num_cores'>
```




```
#fast_charging_available


df['fast_charging_available'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:85.4% of phones in market have fast charging

     <Axes: ylabel='fast_charging_available'>
```

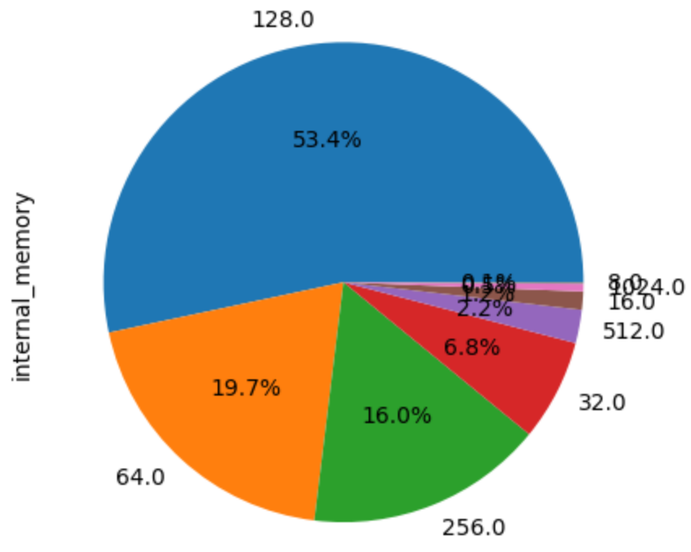



```
#ram_capacity


df['ram_capacity'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:34% of phones have 8gb ram 23% of phones in market have 6gb ram followed by 4 gb which contributes 22.1% of phones in marlet

     <Axes: ylabel='ram_capacity'>
```




```
#internal_memory


df['internal_memory'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion :About 50% of phones in market have 128g internal memory---19% giving 64,16% giving 256gb
```

```
<Axes: ylabel='internal_memory'>
```

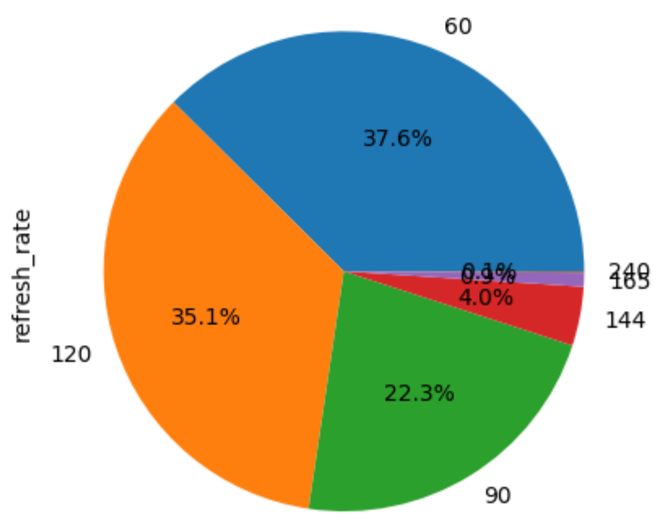



```
#refresh_rate

df['refresh_rate'].describe()
```

```
count    980.000000
mean      92.256122
std       28.988052
min       60.000000
25%       60.000000
50%       90.000000
75%      120.000000
max      240.000000
Name: refresh_rate, dtype: float64
```

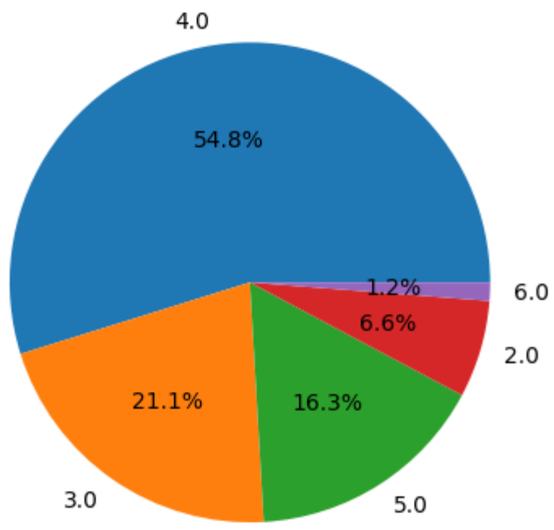
```
df['refresh_rate'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Around 37% phones gies 60hz refresh_rate,35% gives 120hz and 22% gives 90hz
```

```
<Axes: ylabel='refresh_rate'>
```




```
(df['num_rear_cameras']+df['num_front_cameras']).value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Almost 54% of the phones have total 4 cameras---21% have 3 cameras 16.3% have 5 cameras
```

```
<Axes: >
```

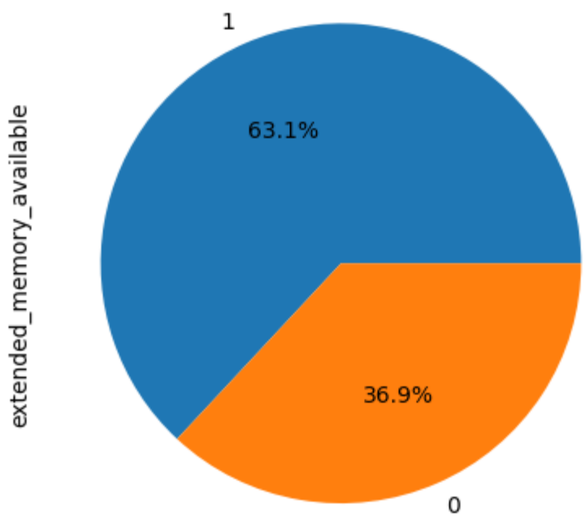



```
df['os'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion: We can see 94% of total phone market is covered by andriod and 4.8% is covered by ios Android is clear dominator in os
```

```
<Axes: ylabel='os'>
```

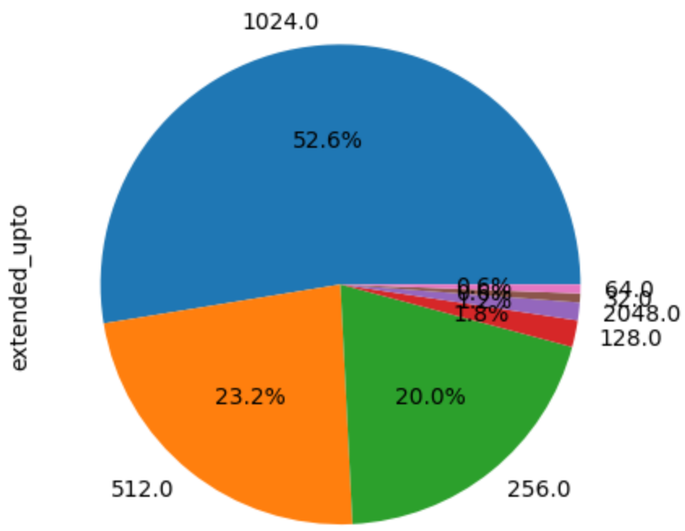
```
df['extended_memory_available'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Around 63% of phones supports extended memory and 36% doesnt
```

```
<Axes: ylabel='extended_memory_available'>
```




```
df['extended_upto'].value_counts().plot(kind='pie',autopct='%0.1f%%')
#Conclusion:Around 52% of phones supports upto 1tb extended memory 23% supports 512gb extended memory and 20% supports 256gb extended memory
```

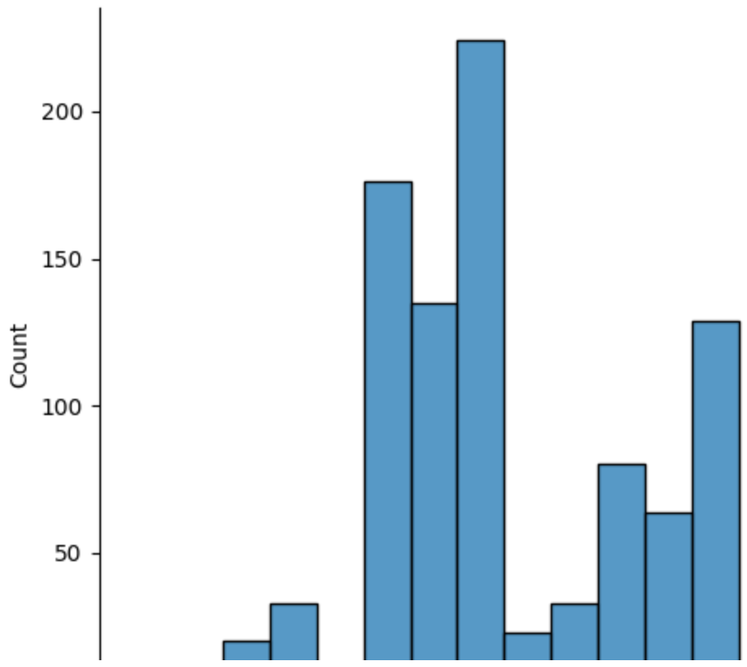
```
<Axes: ylabel='extended_upto'>
```




```
#Univarient Analysis On Numerical Columns

def plot_graphs(col_name):
  sns.displot(kind='hist',data=df,x=col_name)
  sns.catplot(kind='box',data=df,x=col_name)

#Will select only the columns with int and float datatypes
df.select_dtypes(include=['int','float']).head()
```

	price	rating	num_cores	processor_speed	battery_capacity	fast_charging_ava
0	54999	89.0	8.0	3.2	5000.0	
1	19989	81.0	8.0	2.2	5000.0	
2	16499	75.0	8.0	2.4	5000.0	
3	14999	81.0	8.0	2.2	5000.0	
4	24999	82.0	8.0	2.6	5000.0	

```
#Extracting Columns name so that we can provide it to our plot_graphs function
col=df.select_dtypes(include=['int','float']).iloc[:,[3,4,6,9,13,14,16]].columns

for i in col:
  plot_graphs(i)
```



BiVariant **Analysis**

Price-Brand

```
plt.figure(figsize=(20,10))
sns.barplot(data=df,x='brand_name',y='price')
plt.xticks(rotation='vertical')
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45]),
 [Text(0, 0, 'oneplus'),
  Text(1, 0, 'samsung'),
  Text(2, 0, 'motorola'),
  Text(3, 0, 'realme'),
  Text(4, 0, 'apple'),
  Text(5, 0, 'xiaomi'),
  Text(6, 0, 'nothing'),
  Text(7, 0, 'oppo'),
  Text(8, 0, 'vivo'),
  Text(9, 0, 'poco'),
  Text(10, 0, 'iqoo'),
  Text(11, 0, 'jio'),
  Text(12, 0, 'gionee'),
  Text(13, 0, 'tecno'),
  Text(14, 0, 'tesla'),
  Text(15, 0, 'google'),
  Text(16, 0, 'infinix'),
  Text(17, 0, 'cola'),
  Text(18, 0, 'letv'),
  Text(19, 0, 'ikall'),
  Text(20, 0, 'leeco'),
  Text(21, 0, 'duoqin'),
  Text(22, 0, 'nokia'),
  Text(23, 0, 'lava'),
  Text(24, 0, 'honor'),
  Text(25, 0, 'nubia'),
  Text(26, 0, 'redmi'),
  Text(27, 0, 'asus'),
  Text(28, 0, 'itel'),
  Text(29, 0, 'royole'),
  Text(30, 0, 'sony'),
  Text(31, 0, 'oukitel'),
  Text(32, 0, 'vertu'),
  Text(33, 0, 'blu'),
  Text(34, 0, 'lyf'),
  Text(35, 0, 'huawei'),
  Text(36, 0, 'zte'),
  Text(37, 0, 'lenovo'),
  Text(38, 0, 'lg'),
  Text(39, 0, 'micromax'),
  Text(40, 0, 'leitz'),
  Text(41, 0, 'cat'),
  Text(42, 0, 'doogee'),
  Text(43, 0, 'tcl'),
  Text(44, 0, 'sharp'),
  Text(45, 0, 'blackview')])
```


```
#taking only brands which has atleast 10 phones
```

```
x=df.groupby('brand_name').count()['model']
x[x>10].index #taking only those brand which have atleast 10phones
```

```
Index(['apple', 'google', 'honor', 'huawei', 'infinix', 'iqoo', 'motorola',
       'nokia', 'oneplus', 'oppo', 'poco', 'realme', 'samsung', 'tecno',
       'vivo', 'xiaomi'],
      dtype='object', name='brand_name')
```

```
#Sorting values wrt mean price
df[df['brand_name'].isin(x[x>10].index)].groupby('brand_name').mean()['price'].sort_values(ascending=False).index

#taking only brands which have atleast 10 models
x = df.groupby('brand_name').count()['model']
temp_df = df[df['brand_name'].isin(x[x > 10].index)]
```

```
<ipython-input-186-b12629517d35>:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False.
  df[df['brand_name'].isin(x[x>10].index)].groupby('brand_name').mean()['price'].sort_values(ascending=False).index
```

```
#Sorted brand name with respect to their avg price for plotting beautiful barplot
temp=df
temp['brand_name']=pd.Categorical(df['brand_name'], categories=ordered_index, ordered=True)
df_sorted = df.sort_values(by='brand_name')
df_sorted = df.dropna(subset=['brand_name'])
df_sorted['brand_name']=pd.Categorical(df_sorted['brand_name'], categories=ordered_index, ordered=True)
df_sorted = df_sorted.sort_values(by='brand_name')
df_sorted
```

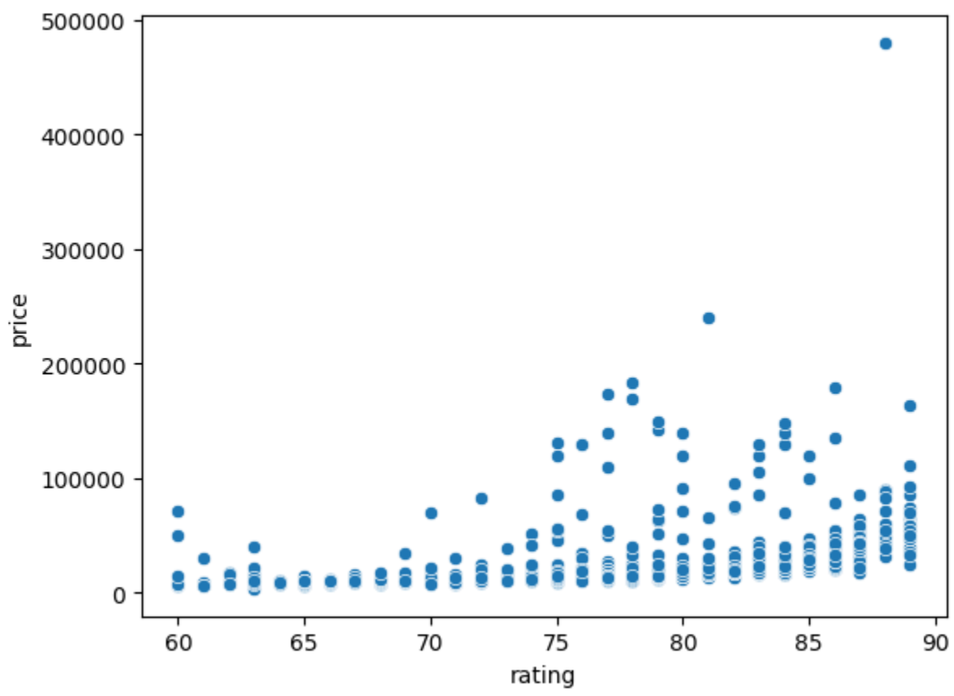
```
#Top 10 brands with respect to avg price
plt.figure(figsize=[10,10])
sns.barplot(data=df_sorted,x='brand_name',y='price')
plt.xticks(rotation='vertical')
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),
 [Text(0, 0, 'apple'),
  Text(1, 0, 'huawei'),
  Text(2, 0, 'google'),
  Text(3, 0, 'samsung'),
  Text(4, 0, 'oneplus'),
  Text(5, 0, 'iqoo')
```

Price-Rating

```
sns.scatterplot(temp_df,x='rating',y='price')
#conclusion=rating is not influencing price at all
```

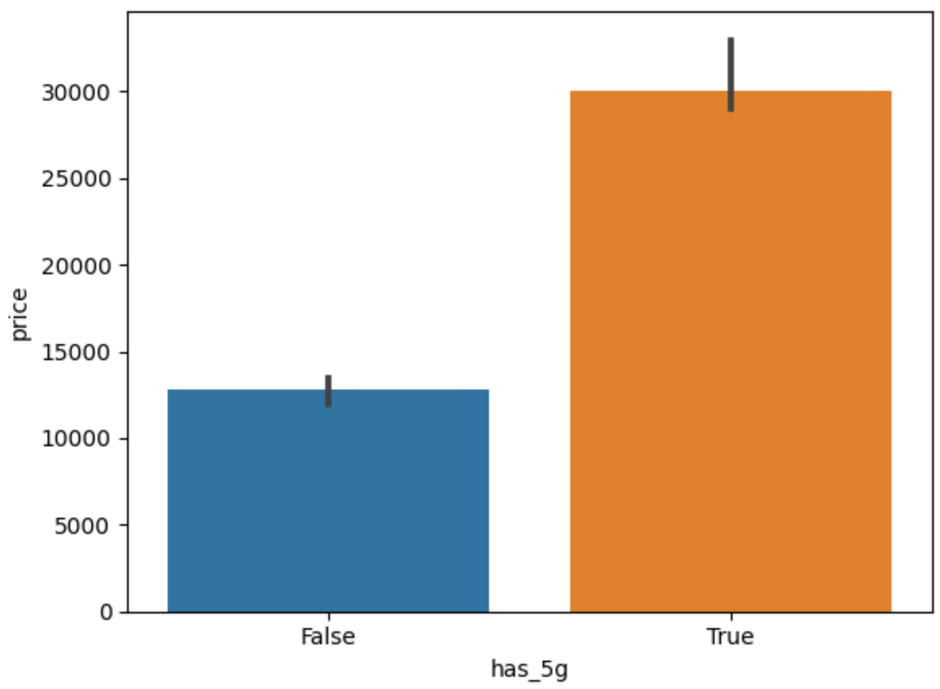
```
<Axes: xlabel='rating', ylabel='price'>
```



has_5g-price

```
sns.barplot(data=df,x='has_5g',y='price',estimator=np.median)
#Conclusion: 5g phones tends to have higher price
```

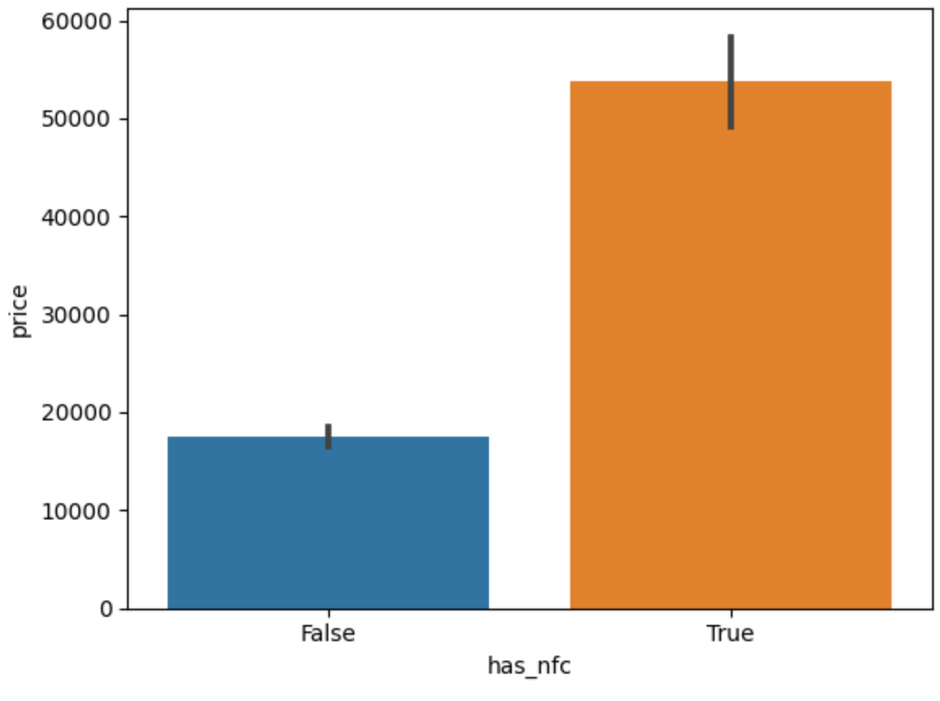
```
<Axes: xlabel='has_5g', ylabel='price'>
```



has_nfc--price

```
sns.barplot(data=temp_df,x='has_nfc',y='price')
#Conclusion:phones which have nfc have higher price,has_nfc column influences price
```

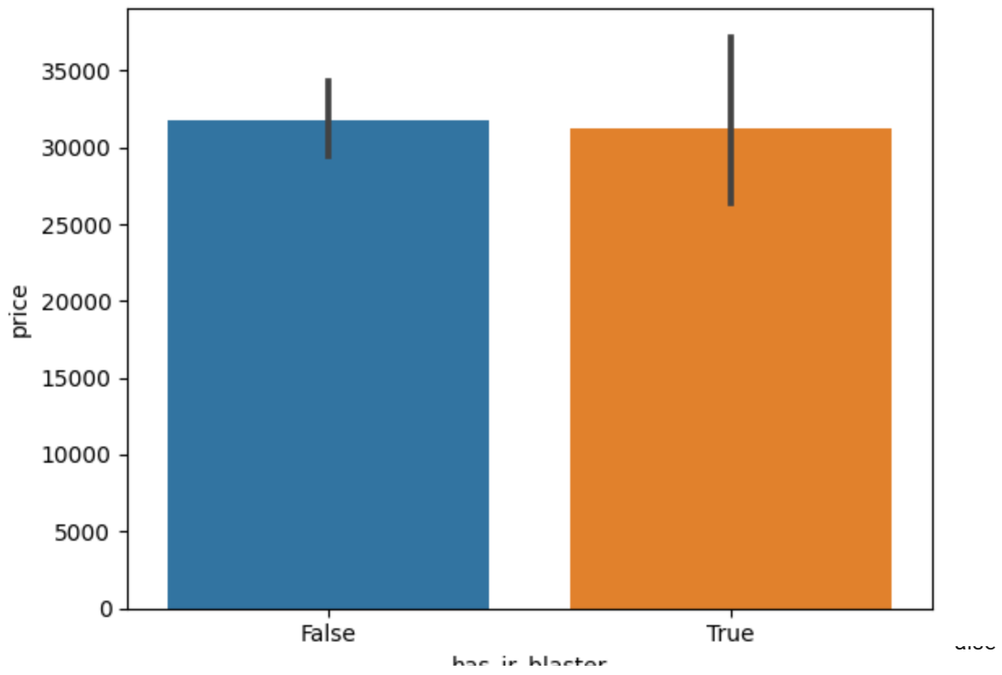
```
<Axes: xlabel='has_nfc', ylabel='price'>
```



IR-Blaster-Price

```
sns.barplot(data=temp_df,x='has_ir_blaster',y='price')
#Conclusion:ir blaster does not influences price
```

```
<Axes: xlabel='has_ir_blaster', ylabel='price'>
```

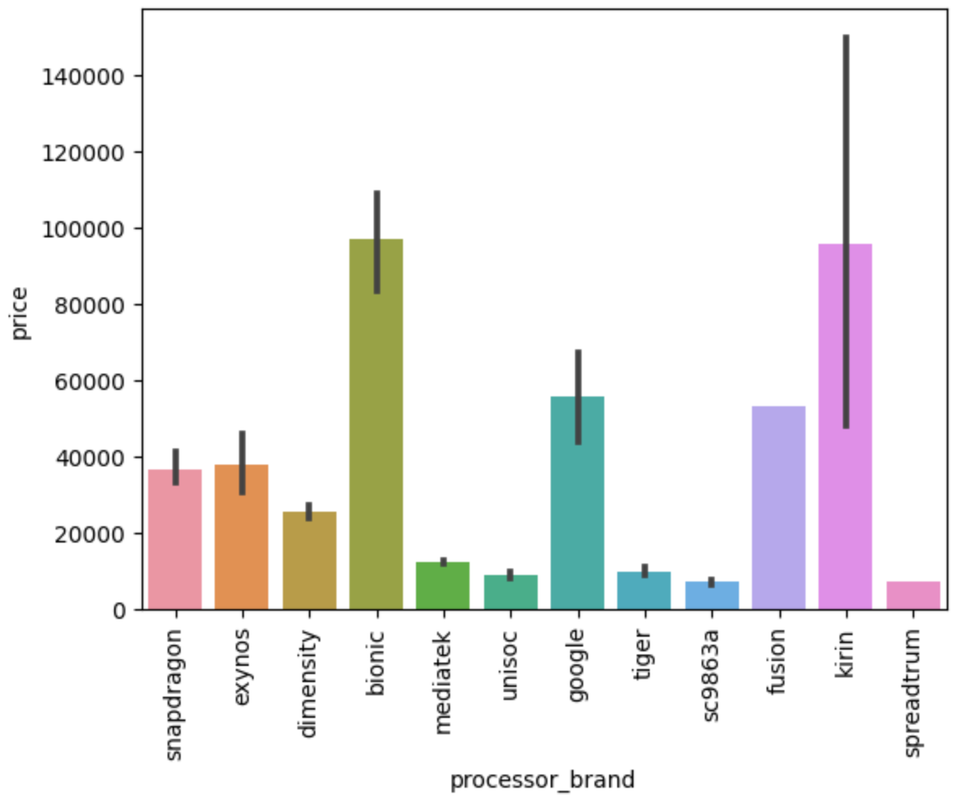


Processor_brand--Price

(1TB)

```
sns.barplot(data=temp_df,x='processor_brand',y='price')
plt.xticks(rotation='vertical')

#Conclusion:phones with bionic processor have highest avg price followed by kirin
```

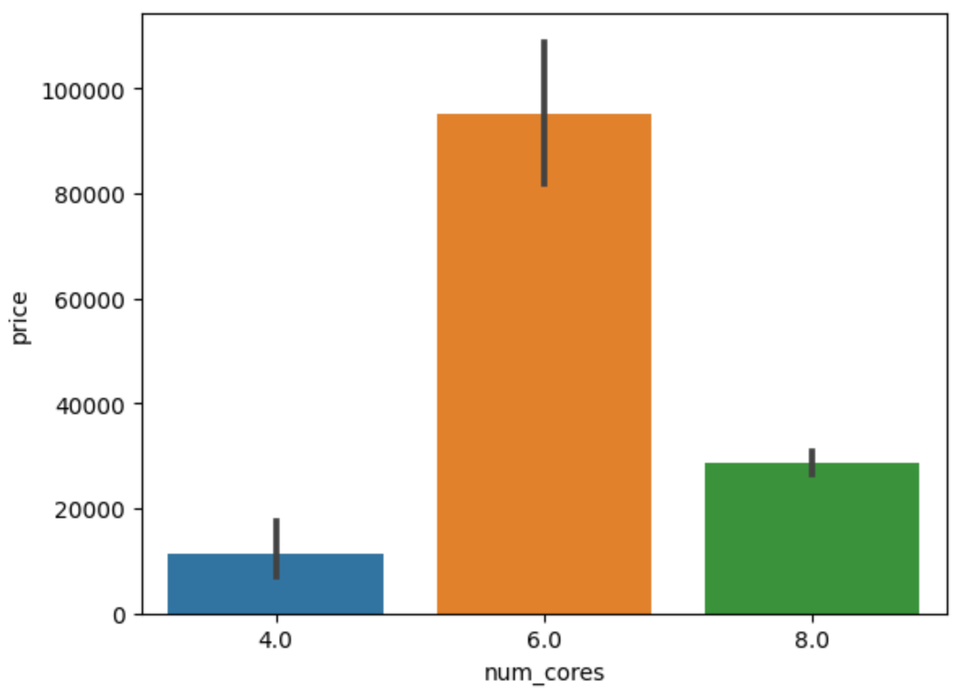
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
 [Text(0, 0, 'snapdragon'),
  Text(1, 0, 'exynos'),
  Text(2, 0, 'dimensity'),
  Text(3, 0, 'bionic'),
  Text(4, 0, 'mediatek'),
  Text(5, 0, 'unisoc'),
  Text(6, 0, 'google'),
  Text(7, 0, 'tiger'),
  Text(8, 0, 'sc9863a'),
  Text(9, 0, 'fusion'),
  Text(10, 0, 'kirin'),
  Text(11, 0, 'spreadtrum')])
```



Huawei

```
sns.barplot(data=temp_df,x='num_cores',y='price')
#Conclusion:4--quad,6--hexa core,8--quad core
''' Devices with hexa core are most costly as hexa core is only used in apple phones'''
```

```
<Axes: xlabel='num_cores', ylabel='price'>
```



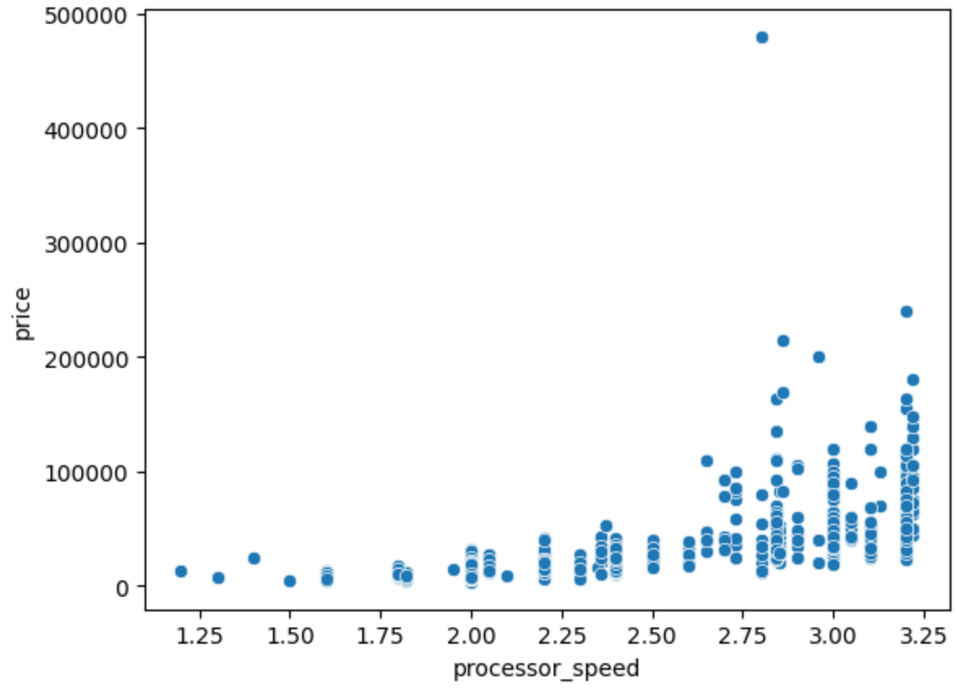
8 Pro

```
pd.crosstab(df['num_cores'],df['os'])
```

os	android	ios	other
num_cores			
4.0	33	1	1
6.0	0	39	0
8.0	875	1	10

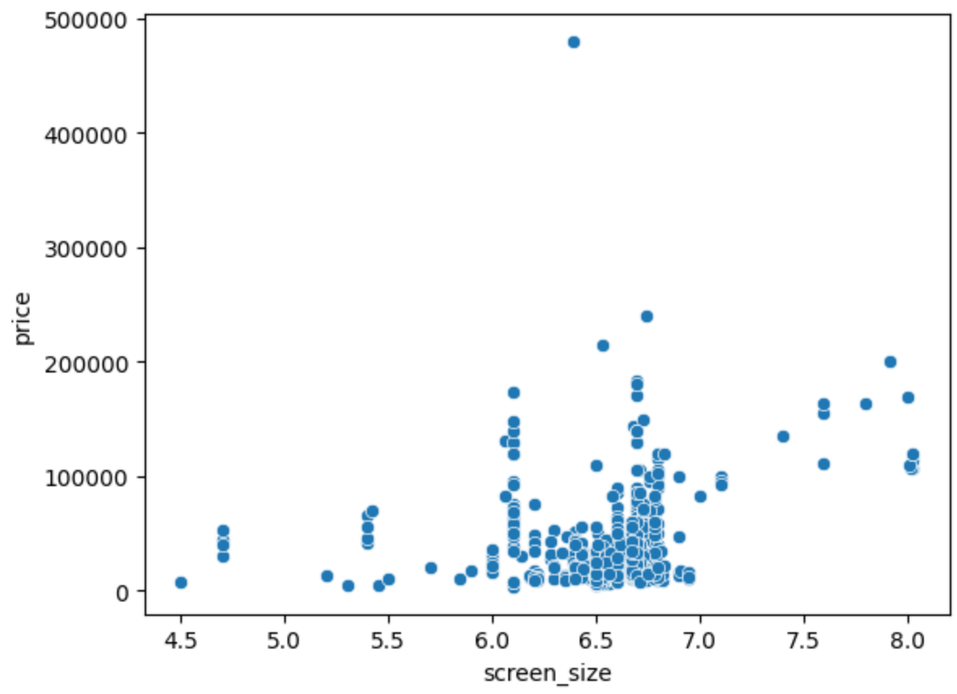
```
sns.scatterplot(data=temp_df,x='processor_speed',y='price')
#Conclusion: higher processor speed in correlated with price
```

```
<Axes: xlabel='processor_speed', ylabel='price'>
```




```
sns.scatterplot(data=temp_df,x='screen_size',y='price')
#Conclusion: Screen_size is not affecting price that much
```

```
<Axes: xlabel='screen_size', ylabel='price'>
```



469	samsung	Galaxy M32	12364	76.0	False	False	False

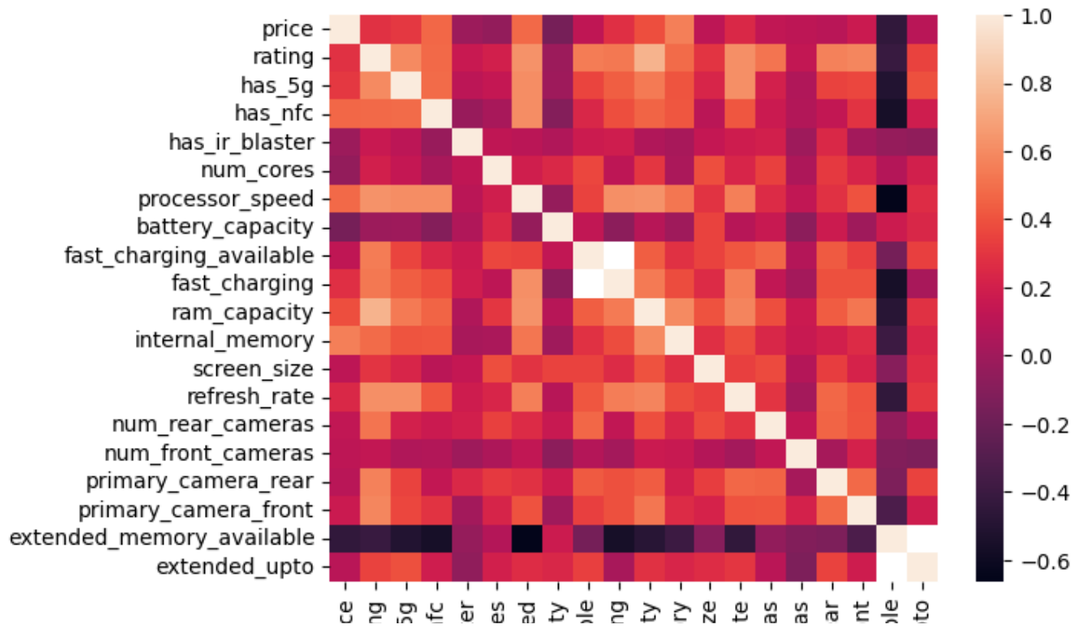
```
#Checking correlation of columns with other columns
temp.corr()
```

```
<ipython-input-200-79cec36d1758>:1: FutureWarning: The default value of numeric_
  temp.corr()
```

	price	rating	has_5g	has_nfc	has_ir_blaster	num_
price	1.000000	0.283504	0.305066	0.470951	-0.015807	-0.0
rating	0.283504	1.000000	0.596087	0.474754	0.156421	0.1
has_5g	0.305066	0.596087	1.000000	0.481702	0.105542	0.1
has_nfc	0.470951	0.474754	0.481702	1.000000	-0.032541	0.0
has_ir_blaster	-0.015807	0.156421	0.105542	-0.032541	1.000000	0.1
num_cores	-0.048561	0.199741	0.139607	0.026165	0.120363	1.0
processor_speed	0.474049	0.628446	0.609583	0.609664	0.102744	0.1
battery_capacity	-0.159232	-0.015581	-0.013237	-0.106104	0.059852	0.2
fast_charging_available	0.116739	0.542814	0.355858	0.237947	0.174060	0.3
fast_charging	0.277591	0.527613	0.440624	0.383231	0.187605	0.1
ram_capacity	0.386002	0.757613	0.533957	0.458336	0.059460	0.3
internal_memory	0.557168	0.481070	0.403837	0.413071	0.030789	0.0
screen_size	0.113253	0.298272	0.230598	0.103099	0.140809	0.3
refresh_rate	0.244115	0.610795	0.611794	0.410777	0.178378	0.2
num_rear_cameras	0.125330	0.515531	0.206512	0.166299	0.198043	0.3
num_front_cameras	0.115228	0.131480	0.058059	0.066278	-0.011380	0.0
primary_camera_rear	0.092095	0.562046	0.347918	0.131004	0.243608	0.3
primary_camera_front	0.162995	0.577861	0.358769	0.285427	0.010399	0.2
extended_memory_available	-0.448628	-0.415265	-0.507752	-0.564380	-0.041676	0.0
extended_upto	0.091945	0.346761	0.392268	0.187599	-0.060974	0.1

```
sns.heatmap(temp.corr())
```

```
<ipython-input-202-0b84c171a535>:1: FutureWarning: The default value of numeric_
  sns.heatmap(temp.corr())
<Axes: >
```




```
#Conclusion :
'''
extended_memory --- processor_speed
as processor speed increases extendend memory decreasess by -0.6 this is because higher proccessor speed is provided mostly in flagship phones
and flagship phones do not provide support for extended memory support
'''
```

```
df.corr()['price']
```

```
<ipython-input-204-cbe57b8e6d9c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only vali
  df.corr()['price']
price                        1.000000
rating                       0.283504
has_5g                       0.305066
has_nfc                      0.470951
has_ir_blaster              -0.015807
num_cores                   -0.048561
processor_speed              0.474049
battery_capacity            -0.159232
fast_charging_available      0.116739
fast_charging                0.277591
ram_capacity                 0.386002
internal_memory              0.557168
screen_size                  0.113253
refresh_rate                 0.244115
num_rear_cameras             0.125330
num_front_cameras            0.115228
primary_camera_rear          0.092095
primary_camera_front         0.162995
extended_memory_available   -0.448628
extended_upto                0.091945
Name: price, dtype: float64
```

428	samsung	Prime	14859	79.0	False	False	False
183	samsung	Samsung Galaxy F24 5G	14999	78.0	True	True	False
400	samsung	Samsung Galaxy Note 30 Ultra 5G	104999	NaN	True	True	False
394	samsung	Samsung Galaxy A13 5G	17990	73.0	True	True	False
390	samsung	Samsung Galaxy M32 Prime Edition	12120	75.0	False	False	False
191	samsung	Samsung Galaxy M54 5G	34999	85.0	True	False	False