In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
df= pd.read_csv('Flipkart Mobile 2.csv')
```

In [3]:
```
df
```

Out[3]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	
...	...	...	...	...	...	...	...	...	
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

430 rows × 16 columns

In [4]:
```
df.head()
```

Out[4]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_came
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	

In [5]:
```
df.tail()
```

Out[5]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

In [6]:
```
df.shape
```

Out[6]:
```
(430, 16)
```

In [7]:
```
df.size
```

Out[7]:
```
6880
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 430 entries, 0 to 429
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   brand             430 non-null    object 
 1   model             430 non-null    object 
 2   base_color        430 non-null    object 
 3   processor         430 non-null    object 
 4   screen_size       430 non-null    object 
 5   ROM               430 non-null    int64  
 6   RAM               430 non-null    int64  
 7   display_size      430 non-null    float64
 8   num_rear_camera   430 non-null    int64  
 9   num_front_camera  430 non-null    int64  
 10  battery_capacity  430 non-null    int64  
 11  ratings           430 non-null    float64
 12  num_of_ratings    430 non-null    int64  
 13  sales_price       430 non-null    int64  
 14  discount_percent  430 non-null    float64
 15  sales             430 non-null    float64
dtypes: float64(4), int64(7), object(5)
memory usage: 53.9+ KB
```

In [9]:
```
df.describe()
```

Out[9]:

	ROM	RAM	display_size	num_rear_camera	num_front_camera	battery_capa
count	430.000000	430.000000	430.000000	430.000000	430.000000	430.000
mean	105.748837	5.320930	6.369767	2.904651	1.044186	4529.397
std	63.164064	2.182635	0.369549	0.952350	0.227280	986.907
min	8.000000	1.000000	4.700000	1.000000	1.000000	1800.000
25%	64.000000	4.000000	6.300000	2.000000	1.000000	4000.000
50%	128.000000	4.000000	6.500000	3.000000	1.000000	4500.000
75%	128.000000	6.000000	6.500000	4.000000	1.000000	5000.000
max	512.000000	12.000000	7.600000	4.000000	3.000000	7000.000

In [10]:
```
df['ROM'].mean()
```

Out[10]:
```
105.74883720930232
```

how many mobile brands are there?

In [11]:
```
df['brand'].unique()
```

Out[11]:
```
array(['Apple', 'Poco', 'Realme', 'Samsung', 'Xiaomi'], dtype=object)
```

get seperate them all one by one

In [84]:
```
branddf=df.groupby('brand')
```

In [13]:
```
branddf.groups
```

Out[13]:
```
{'Apple': [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, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55], 'Po
co': [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 7
3, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 9
2, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108,
109, 110, 111], 'Realme': [112, 113, 114, 115, 116, 117, 118, 119, 120, 121,
122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 1
37, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 15
2, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 16
7, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 18
2, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 19
7, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211,
...], 'Samsung': [250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 26
1, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 27
6, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 29
1, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 30
6, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 32
1, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 33
6, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, ...], 'X
iaomi': [369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 38
2, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 39
7, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 41
2, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 42
7, 428, 429]}
```

seperate each group

In [14]:
```
appledf=branddf.get_group('Apple')
```

In [15]:
```
appledf
```

Out[15]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	
5	Apple	iPhone 12	Blue	Ceramic	Medium	64	4	6.1	
6	Apple	iPhone	White	Ceramic	Medium	128	4	6.1	

In [16]:
```
pocodf=branddf.get_group('Poco')
```

In [17]:
```
pocodf
```

Out[17]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_
56	Poco	C3	Black	MediaTek	Large	64	4	6.5	
57	Poco	M3	Blue	Qualcomm	Large	64	4	6.5	
58	Poco	M2 Reloaded	Blue	MediaTek	Large	64	4	6.5	
59	Poco	C3	Blue	MediaTek	Large	32	3	6.5	
60	Poco	M3	Black	Qualcomm	Large	64	6	6.5	
61	Poco	C3	Black	MediaTek	Large	32	3	6.5	
62	Poco	M3	Blue	Qualcomm	Large	64	6	6.5	
63	Poco	M2 Reloaded	Black	MediaTek	Large	64	4	6.5	
64	Poco	M3	Yellow	Qualcomm	Large	64	6	6.5	
65	Poco	M3	Yellow	Qualcomm	Large	64	4	6.5	
66	Poco	M3	Black	Qualcomm	Large	64	4	6.5	
67	Poco	M3	Black	Qualcomm	Large	128	6	6.5	
68	Poco	C3	Blue	MediaTek	Large	64	4	6.5	
69	Poco	C3	Green	MediaTek	Large	32	3	6.5	
70	Poco	C3	Green	MediaTek	Large	64	4	6.5	
71	Poco	X3 Pro	Black	Qualcomm	Large	128	6	6.7	
72	Poco	M3 Pro 5G	Yellow	MediaTek	Large	128	6	6.5	
73	Poco	X3 Pro	Blue	Qualcomm	Large	128	6	6.7	
74	Poco	X3 Pro	Black	Qualcomm	Large	128	8	6.7	
75	Poco	X3 Pro	Blue	Qualcomm	Large	128	8	6.7	
76	Poco	M3 Pro 5G	Blue	MediaTek	Large	64	4	6.5	
77	Poco	M3 Pro 5G	Blue	MediaTek	Large	128	6	6.5	
78	Poco	M2 Pro	Others	Qualcomm	Large	128	6	6.7	
79	Poco	X3 Pro	Bronze	Qualcomm	Large	128	6	6.7	
80	Poco	M3 Pro 5G	Black	MediaTek	Large	128	6	6.5	
81	Poco	M2 Pro	Black	Qualcomm	Large	128	6	6.7	
82	Poco	M2 Pro	Blue	Qualcomm	Large	128	6	6.7	
83	Poco	X3 Pro	Bronze	Qualcomm	Large	128	8	6.7	
84	Poco	M3	Blue	Qualcomm	Large	128	6	6.5	
85	Poco	M3 Pro 5G	Yellow	MediaTek	Large	64	4	6.5	
86	Poco	F3 GT	Black	MediaTek	Large	128	8	6.7	
87	Poco	M2 Pro	Blue	Qualcomm	Large	64	6	6.7	

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_
88	Poco	M2 Pro	Blue	Qualcomm	Large	64	4	6.7	
89	Poco	M2 Pro	Others	Qualcomm	Large	64	4	6.7	
90	Poco	M2 Pro	Black	Qualcomm	Large	64	4	6.7	
91	Poco	X3	Blue	Qualcomm	Large	128	8	6.7	
92	Poco	F3 GT	Silver	MediaTek	Large	256	8	6.7	
93	Poco	F3 GT	Black	MediaTek	Large	256	8	6.7	
94	Poco	F3 GT	Silver	MediaTek	Large	128	8	6.7	
95	Poco	F3 GT	Black	MediaTek	Large	128	6	6.7	
96	Poco	F3 GT	Silver	MediaTek	Large	128	6	6.7	
97	Poco	F1	Blue	Qualcomm	Medium	256	8	6.2	
98	Poco	F1	Black	Qualcomm	Medium	128	6	6.2	
99	Poco	M3	Yellow	Qualcomm	Large	128	6	6.5	
100	Poco	M2 Pro	Black	Qualcomm	Large	64	6	6.7	
101	Poco	M2 Pro	Others	Qualcomm	Large	64	6	6.7	
102	Poco	M2	Red	MediaTek	Large	128	6	6.5	
103	Poco	X3	Blue	Qualcomm	Large	128	6	6.7	
104	Poco	X3	Blue	Qualcomm	Large	64	6	6.7	
105	Poco	X3	Gray	Qualcomm	Large	64	6	6.7	
106	Poco	X3	Gray	Qualcomm	Large	128	6	6.7	
107	Poco	M2	Black	MediaTek	Large	128	6	6.5	
108	Poco	M2	Blue	MediaTek	Large	128	6	6.5	
109	Poco	X3	Gray	Qualcomm	Large	128	8	6.7	
110	Poco	X2	Red	Qualcomm	Large	128	6	6.7	
111	Poco	M2	Red	MediaTek	Large	64	6	6.5	

In [18]:
```
realmedf=branddf.get_group('Realme')
```

In [19]:
```
realmedf
```

Out[19]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_c
112	Realme	C20	Blue	MediaTek	Large	32	2	6.5	
113	Realme	C20	Gray	MediaTek	Large	32	2	6.5	
114	Realme	C11 2021	Gray	Others	Large	32	2	6.5	
115	Realme	C11 2021	Blue	Others	Large	32	2	6.5	
116	Realme	C21Y	Black	Others	Large	64	4	6.5	
...	...	...	...	...	...	...	...	...	
245	Realme	Narzo 10A	Blue	MediaTek	Large	32	3	6.5	
246	Realme	Narzo 20A	Silver	Qualcomm	Large	32	3	6.5	
247	Realme	Narzo 10	Green	MediaTek	Large	128	4	6.5	
248	Realme	Narzo 10	White	MediaTek	Large	128	4	6.5	
249	Realme	Narzo 20A	Silver	Qualcomm	Large	64	4	6.5	

138 rows × 16 columns

In [20]:
```
xiaomidf= branddf.get_group('Xiaomi')
```

In [21]:
```
xiaomidf
```

Out[21]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
369	Xiaomi	Redmi 9A	Black	MediaTek	Large	32	3	6.5	
370	Xiaomi	Mi 11X	White	Qualcomm	Large	128	6	6.7	
371	Xiaomi	Redmi 8A Dual	White	Qualcomm	Medium	32	3	6.2	
372	Xiaomi	Mi A3	Blue	Qualcomm	Medium	64	4	6.1	
373	Xiaomi	Redmi 9	Blue	MediaTek	Large	128	4	6.5	
...	...	...	...	...	...	...	...	...	
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

61 rows × 16 columns

In [22]:
```
samsungdf=branddf.get_group('Samsung')
samsungdf
```

Out[22]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear
250	Samsung	Galaxy F22	Black	MediaTek	Medium	64	4	6.4	
251	Samsung	Galaxy F22	Blue	MediaTek	Medium	64	4	6.4	
252	Samsung	Galaxy F22	Blue	MediaTek	Medium	128	6	6.4	
253	Samsung	Galaxy F22	Black	MediaTek	Medium	128	6	6.4	
254	Samsung	Galaxy F12	Blue	Exynos	Large	64	4	6.5	
...	...	...	...	...	...	...	...	...	
364	Samsung	Galaxy A51	Blue	Exynos	Large	128	6	6.5	
365	Samsung	Galaxy A20s	Black	Qualcomm	Large	32	3	6.5	
366	Samsung	Galaxy A20s	Green	Qualcomm	Large	64	4	6.5	
367	Samsung	Galaxy S10	Black	Exynos	Medium	128	8	6.1	
368	Samsung	Galaxy A20s	Blue	Qualcomm	Large	64	4	6.5	

119 rows × 16 columns

how much is count for each brand

In [23]:
```
df['brand'].value_counts()
```

Out[23]:
```
Realme     138
Samsung    119
Xiaomi      61
Apple       56
Poco        56
Name: brand, dtype: int64
```

find out how many models having each brand

In [24]:
```
appledf['model'].value_counts()
```

Out[24]:
```
iPhone XR         18
iPhone 12         17
iPhone 12 Mini    16
iPhone SE          3
iPhone 8           1
iPhone 7 Plus      1
Name: model, dtype: int64
```

In [25]:
```
pocodf['model'].value_counts()
```

Out[25]:
```
M3             9
M2 Pro         9
C3             6
X3 Pro         6
F3 GT          6
X3             6
M3 Pro 5G      5
M2             4
M2 Reloaded    2
F1             2
X2             1
Name: model, dtype: int64
```

In [26]:
```
realmedf['model'].value_counts()
```

Out[26]:
```
GT Master Edition    9
X3 SuperZoom         6
7 Pro                6
Narzo 30             6
3i                   6
5 Pro                6
8 5G                 6
8 Pro                6
X7 Max               6
8                    6
X7 5G                4
Narzo 30 Pro 5G      4
C25                  4
C11 2021             4
Narzo 20             4
C2                   4
7                    4
C15                  4
Narzo 30A            4
C21                  4
Narzo 30 5G          4
X3                   3
GT 5G                3
Narzo 10A            3
6                    3
Narzo 20A            2
C20                  2
C11                  2
X7 Pro 5G            2
Narzo 20 Pro         2
C12                  2
8s 5G                2
C21Y                 2
Narzo 10             2
6i                   1
Name: model, dtype: int64
```

In [27]:
```
xiaomidf['model'].value_counts()
```

Out[27]:
```
Redmi 6 Pro          6
Mi 11 Lite           6
Redmi Note 7 Pro     6
Redmi Note 6 Pro     5
Redmi Note 9 Pro     4
Redmi 9              3
Mi A3                3
Mi 10T               3
Redmi Y3             3
Mi 10i               2
Redmi 8A Dual        2
Redmi Note 5 Pro     2
Redmi Note 7         2
Mi 11X               2
Redmi K20            2
Mi 10                1
Redmi 6A             1
Mi A1                1
Mi 11X Pro 5G        1
Redmi 9A             1
Redmi Y2             1
Mi 10T Pro           1
Redmi Note 5         1
Redmi Note 4         1
Redmi 5              1
Name: model, dtype: int64
```

In [28]:
```
samsungdf['model'].value_counts()
```

Out[28]:
```
Galaxy A21s                7
Galaxy F62                 6
Galaxy F12                 6
Galaxy F41                 6
Galaxy A03s                6
Galaxy A20s                5
Galaxy F02s                5
Galaxy A51                 5
Galaxy A12                 5
Galaxy F22                 4
Galaxy M02                 4
Galaxy A52s 5G             4
Galaxy Z Flip3 5G          4
Galaxy A22 5G              4
Galaxy A52                 3
Galaxy S20 FE              3
Galaxy Z Fold3 5G          3
Galaxy Note 20             3
Galaxy A50s                2
Galaxy Note 20 Ultra 5G    2
Galaxy M01                 2
Galaxy Grand 2             2
Galaxy J7 - 6              2
Galaxy A71                 2
Galaxy A72                 2
Galaxy M31                 2
Galaxy M32                 2
M02s                       2
Galaxy A31                 2
Galaxy M11                 1
Galaxy S21 Plus            1
Galaxy S21                 1
Galaxy Note10 Lite         1
Galaxy M30s                1
Galaxy M42                 1
Galaxy A10                 1
Galaxy A80                 1
Galaxy A20                 1
Galaxy A22                 1
Galaxy J6                  1
Galaxy A7                  1
Galaxy Fold 2              1
Galaxy S10                 1
Name: model, dtype: int64
```

find out unique model for each brand

In [29]:
```
appledf['model'].unique()
```

Out[29]:
```
array(['iPhone SE', 'iPhone 12 Mini', 'iPhone XR', 'iPhone 12',
       'iPhone 8', 'iPhone 7 Plus'], dtype=object)
```

In [30]:
```
appledf['model'].nunique()
```

Out[30]:
```
6
```

In [31]:
```
samsungdf['model'].unique()
```

Out[31]:
```
array(['Galaxy F22', 'Galaxy F12', 'M02s', 'Galaxy M02', 'Galaxy A22',
       'Galaxy A52s 5G', 'Galaxy M32', 'Galaxy Z Flip3 5G',
       'Galaxy A22 5G', 'Galaxy A21s', 'Galaxy A03s', 'Galaxy M31',
       'Galaxy A51', 'Galaxy A72', 'Galaxy A12', 'Galaxy F62',
       'Galaxy A31', 'Galaxy A52', 'Galaxy F02s', 'Galaxy M11',
       'Galaxy F41', 'Galaxy A71', 'Galaxy Note 20', 'Galaxy Z Fold3 5G',
       'Galaxy M01', 'Galaxy A50s', 'Galaxy Note 20 Ultra 5G',
       'Galaxy S20 FE', 'Galaxy Grand 2', 'Galaxy Fold 2', 'Galaxy A7',
       'Galaxy J6', 'Galaxy J7 - 6', 'Galaxy A10', 'Galaxy A20',
       'Galaxy A80', 'Galaxy S21 Plus', 'Galaxy M42', 'Galaxy M30s',
       'Galaxy A20s', 'Galaxy Note10 Lite', 'Galaxy S21', 'Galaxy S10'],
      dtype=object)
```

In [32]:
```
samsungdf['model'].nunique()
```

Out[32]:
```
43
```

In [33]:
```
xiaomidf['model'].unique()
```

Out[33]:
```
array(['Redmi 9A', 'Mi 11X', 'Redmi 8A Dual', 'Mi A3', 'Redmi 9',
       'Mi 11 Lite', 'Redmi Note 7 Pro', 'Mi 10i', 'Redmi Note 6 Pro',
       'Redmi Note 9 Pro', 'Redmi Note 5 Pro', 'Redmi Note 7', 'Redmi Y3',
       'Redmi K20', 'Redmi Note 4', 'Redmi Note 5', 'Mi 10T',
       'Mi 10T Pro', 'Redmi Y2', 'Mi 10', 'Redmi 6 Pro', 'Mi 11X Pro 5G',
       'Mi A1', 'Redmi 6A', 'Redmi 5'], dtype=object)
```

In [34]:
```
samsungdf['model'].nunique()
```

Out[34]:
```
43
```

In [35]:
```
realmedf['model'].unique()
```

Out[35]:
```
array(['C20', 'C11 2021', 'C21Y', 'Narzo 30 5G', 'C21', 'Narzo 30',
       '8s 5G', 'Narzo 30A', '8 5G', '8 Pro', 'C15', '8',
       'GT Master Edition', 'X7 5G', '7', 'Narzo 30 Pro 5G', 'C12', 'C11',
       'X7 Max', 'GT 5G', '5 Pro', '3i', 'Narzo 20 Pro', '7 Pro',
       'X3 SuperZoom', 'X7 Pro 5G', 'C2', 'X3', '6', '6i', 'C25',
       'Narzo 20', 'Narzo 10A', 'Narzo 20A', 'Narzo 10'], dtype=object)
```

In [36]:
```
realmedf['model'].nunique()
```

Out[36]:
```
35
```

In [37]:
```
df
```

Out[37]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	
...	...	...	...	...	...	...	...	...	
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

430 rows × 16 columns

analysing the sales column

In [38]:
```
df['revenue']= (df['sales']*10000000).round().astype(int)
df.head()
```

Out[38]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_cam
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	

In [39]:
```
df['units_sold']= (df['revenue']/df['sales_price']).round().astype(int)
```

In [40]:
```
df
```

Out[40]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	
...	...	...	...	...	...	...	...	...	
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

430 rows × 18 columns

In [41]:
```
df['units_sold'].sum()
```

Out[41]:
```
7376835
```

In [42]:
```
print('Sales generated in crosess',df.sales.sum(),"cr")
```

```
Sales generated in crosess 12793.5 cr
```

In [43]:
```
df.revenue.sum()/10000000
```

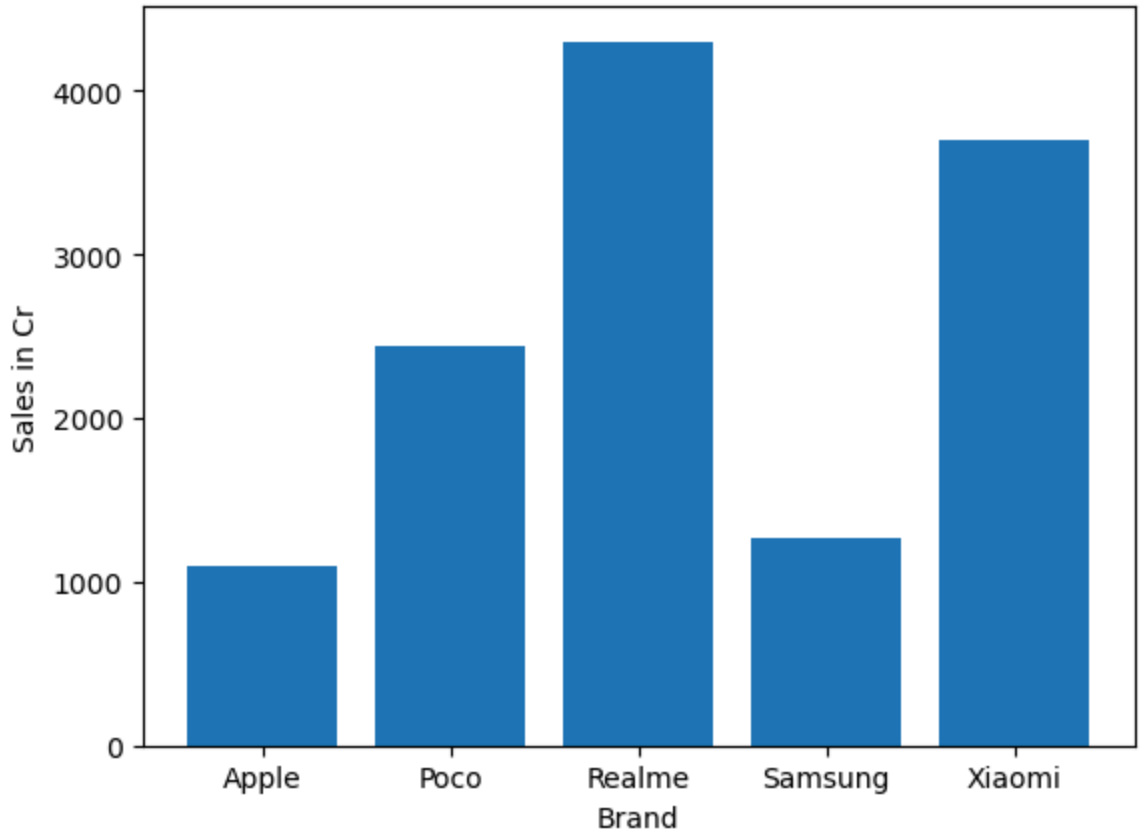
Out[43]:
```
9623.848176
```

In [44]:
```
sales= branddf.sales.sum()
```

In [45]:
```
sales
```

Out[45]:
```
brand
Apple      1091.27
Poco       2437.32
Realme     4301.91
Samsung    1261.90
Xiaomi     3701.10
Name: sales, dtype: float64
```

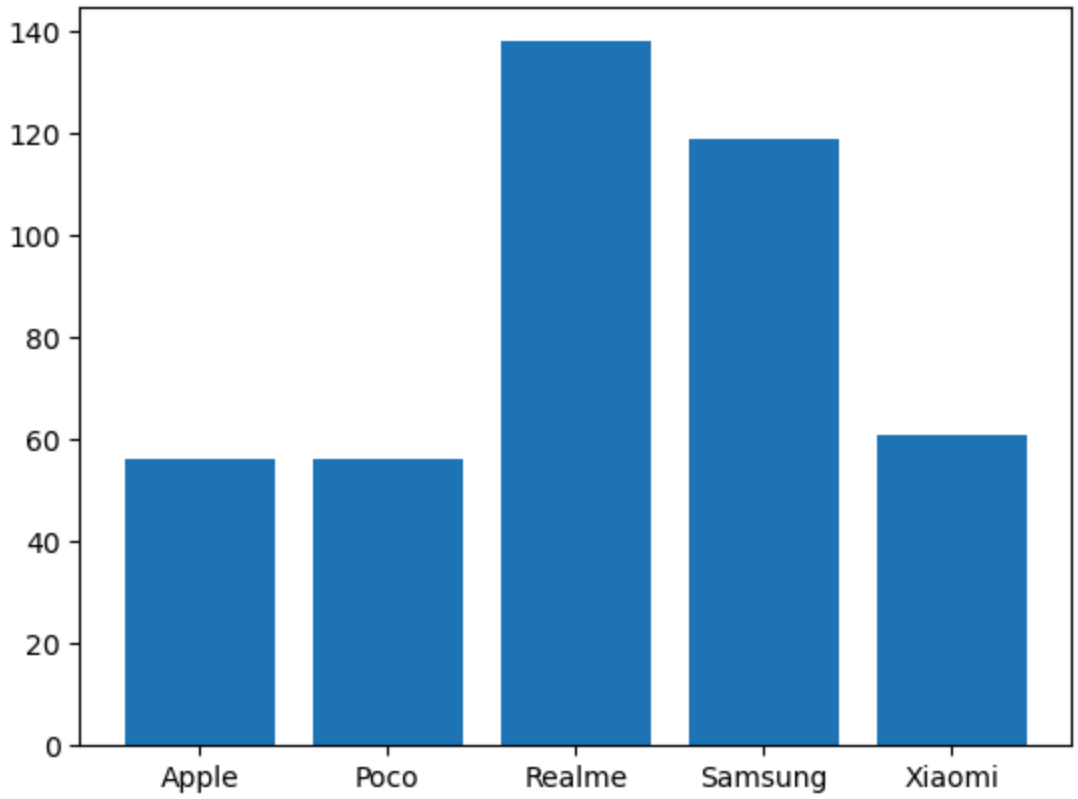
In [46]:
```
plt.bar(sales.index,sales)
plt.xlabel('Brand')
plt.ylabel('Sales in Cr')
plt.show()
```



how many models are sold by each brand

In [47]:
```
sold_model=branddf.model.count()
```

In [48]:
```
plt.bar(sold_model.index,sold_model)
plt.show()
```



what is the average discount given by brands on their model

In [49]:
```
df['actual_discount']= df['sales_price']* df['discount_percent']
```

In [50]:
```
df['actual_discount']
```

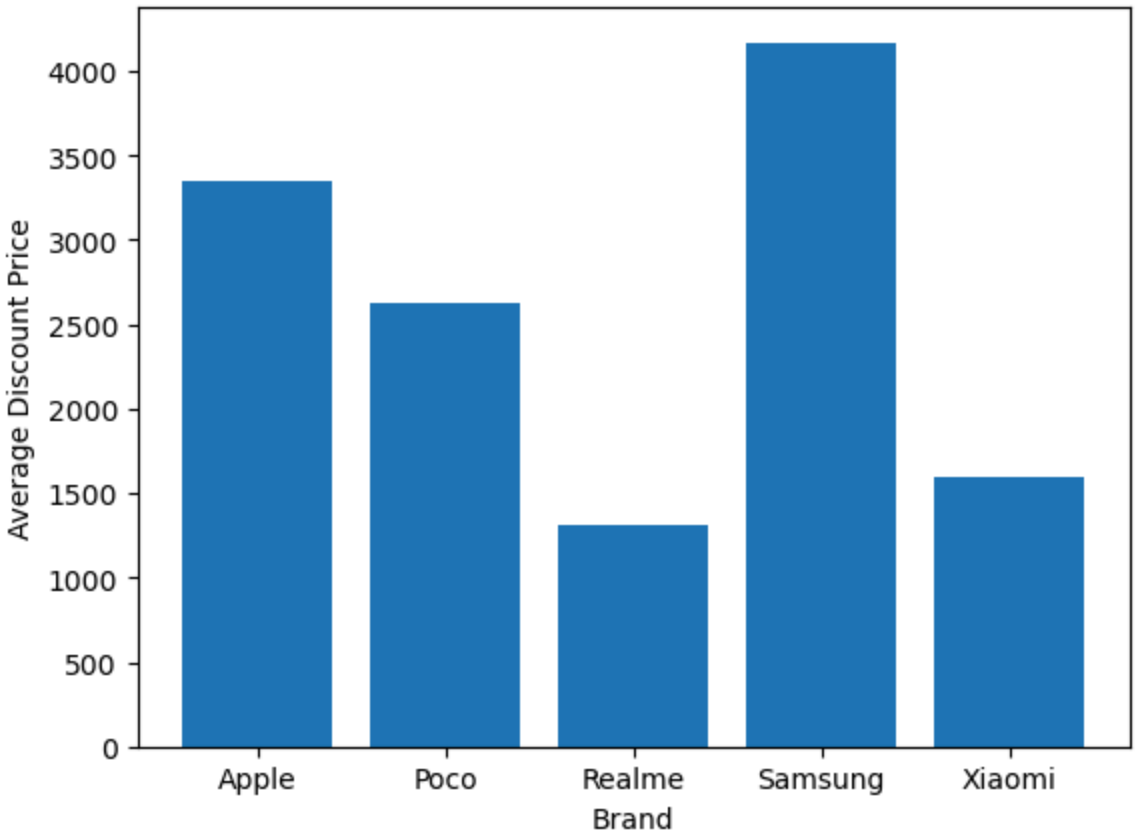
Out[50]:
```
0      5609.83
1      2285.96
2      5609.83
3      4299.90
4      1382.98
        ...
425    2399.70
426    2715.72
427    2639.88
428     580.93
429    2948.40
Name: actual_discount, Length: 430, dtype: float64
```

In [51]:
```
avg_discount= df.groupby('brand').actual_discount.mean()
```

In [52]:
```
avg_discount
```

Out[52]:
```
brand
Apple      3353.346429
Poco       2625.016429
Realme     1315.306884
Samsung    4170.121092
Xiaomi     1600.749180
Name: actual_discount, dtype: float64
```

In [53]:
```
plt.bar(avg_discount.index,avg_discount)
plt.xlabel('Brand')
plt.ylabel('Average Discount Price')
plt.show()
```



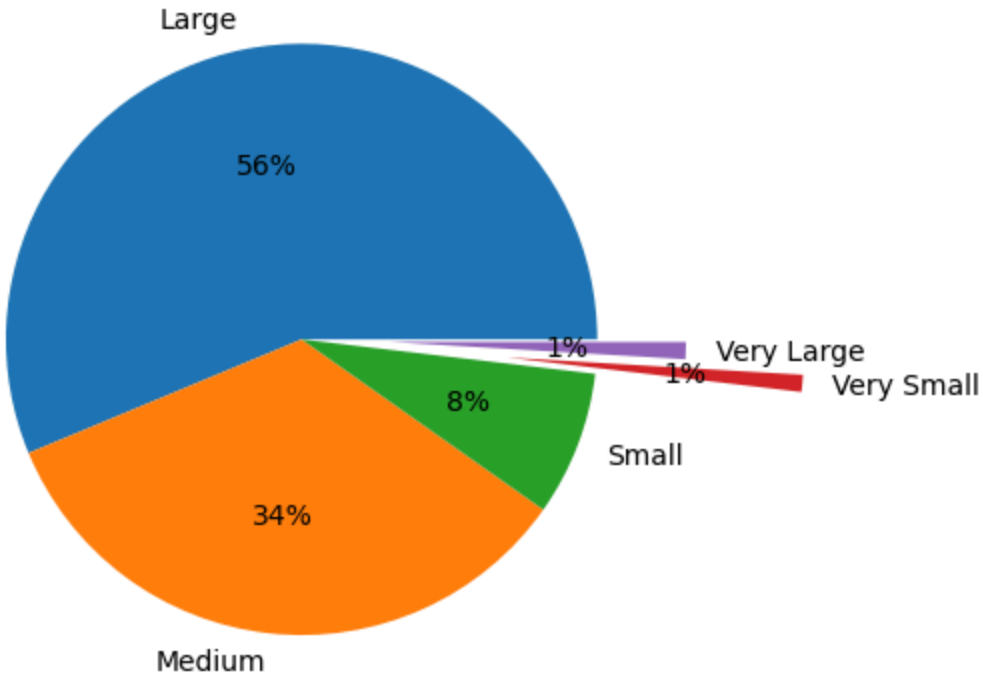
what size of display did customers like the most

In [54]:
```
screen= df.screen_size.value_counts()
```

In [55]:
```
screen
```

Out[55]:
```
Large         242
Medium        146
Small          34
Very Small      4
Very Large      4
Name: screen_size, dtype: int64
```

In [56]:
```
plt.pie(screen, labels= screen.index,autopct='%0.f%%',explode=[0,0,0,0.7,0.3]
plt.show()
```

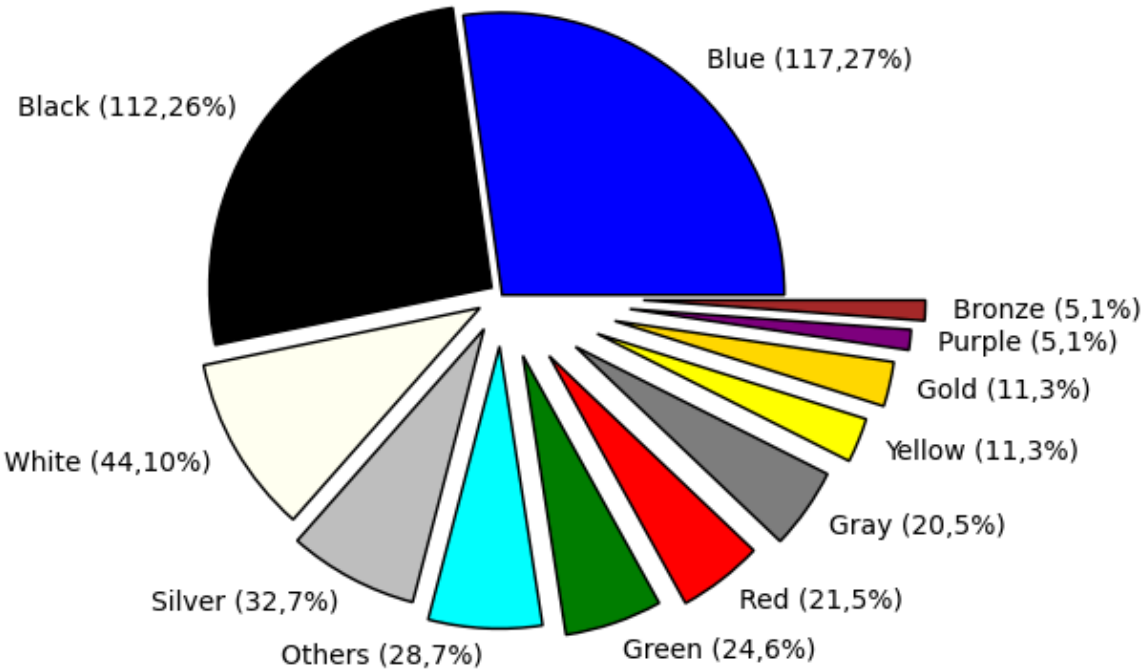


which are favorite colors of mobile phone customers

In [57]:
```
color=df['base_color'].value_counts()
```

In [58]:
```
autopct_color='orange'
combined_labels = [f"{label} ({size},{size/430*100:.0f}%)" for label, size in
c1=['blue','black','ivory','silver','cyan','green','red','grey','yellow','gol
plt.pie(color,labels=combined_labels, colors=c1,explode=np.linspace(0,0.5,12)
plt.show
```

Out[58]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [59]:
```
branddf['base_color'].value_counts()
```

Out[59]:

brand	base_color	
Apple	Black	12
	White	11
	Blue	8
	Red	7
	Green	6
	Others	4
	Purple	4
	Yellow	3
	Gold	1
Poco	Blue	18
	Black	17
	Yellow	5
	Gray	3
	Others	3
	Red	3
	Silver	3
	Bronze	2
	Green	2
Realme	Blue	43
	Black	26
	Silver	24
	White	16
	Gray	9
	Others	9
	Green	5
	Yellow	3
	Red	2
	Purple	1
Samsung	Black	38
	Blue	33
	Green	10
	White	10
	Others	9
	Gray	7
	Red	4
	Bronze	3
	Silver	3
	Gold	2
Xiaomi	Black	19
	Blue	15
	Gold	8
	White	7
	Red	5
	Others	3
	Silver	2
	Gray	1
	Green	1

Name: base_color, dtype: int64

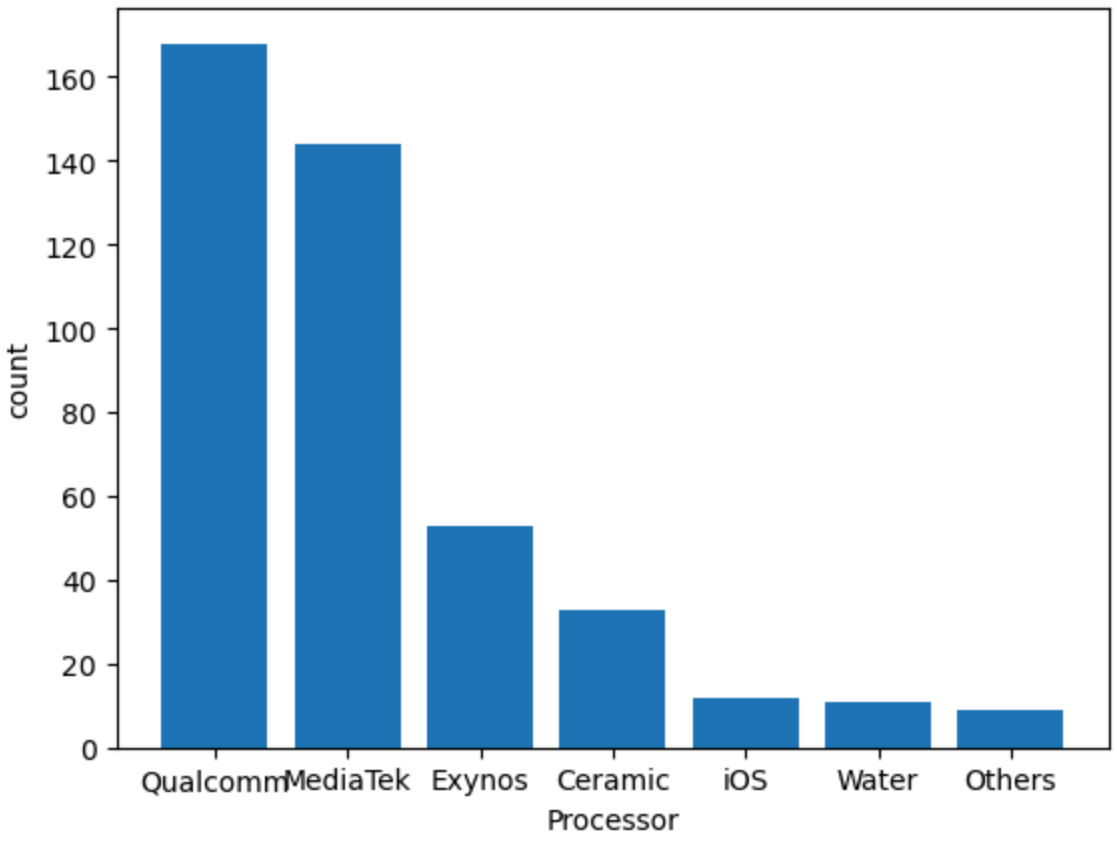
which processors are the most favorable for

In [60]:
```
pro= df['processor'].value_counts()
```

In [61]:
```
pro
```

Out[61]:
```
Qualcomm    168
MediaTek    144
Exynos       53
Ceramic      33
iOS          12
Water        11
Others        9
Name: processor, dtype: int64
```

In [62]:
```
plt.bar(pro.index,pro)
plt.xlabel('Processor')
plt.ylabel('count')
plt.show()
```



In [63]:
```
branddf['processor'].value_counts()
```

Out[63]:
```
brand    processor
Apple    Ceramic      33
         iOS          12
         Water        11
Poco     Qualcomm     33
         MediaTek     23
Realme   MediaTek     91
         Qualcomm     41
         Others        6
Samsung  Exynos       53
         Qualcomm     38
         MediaTek     25
         Others        3
Xiaomi   Qualcomm     56
         MediaTek      5
Name: processor, dtype: int64
```

7. Find out mobile phones at various price ranges. which budget range do customers choose mostly?

In [64]:
```
df
```

Out[64]:

	brand	model	base_color	processor	screen_size	ROM	RAM	display_size	num_rear_ca
0	Apple	iPhone SE	Black	Water	Very Small	64	2	4.7	
1	Apple	iPhone 12 Mini	Red	Ceramic	Small	64	4	5.4	
2	Apple	iPhone SE	Red	Water	Very Small	64	2	4.7	
3	Apple	iPhone XR	Others	iOS	Medium	64	3	6.1	
4	Apple	iPhone 12	Red	Ceramic	Medium	128	4	6.1	
...	...	...	...	...	...	...	...	...	
425	Xiaomi	Redmi 6 Pro	Black	Qualcomm	Small	32	3	5.8	
426	Xiaomi	Redmi 6 Pro	Red	Qualcomm	Small	64	4	5.8	
427	Xiaomi	Mi 11 Lite	Others	Qualcomm	Large	128	6	6.5	
428	Xiaomi	Redmi 8A Dual	Blue	Qualcomm	Medium	32	3	6.2	
429	Xiaomi	Redmi 6 Pro	Blue	Qualcomm	Small	32	3	5.8	

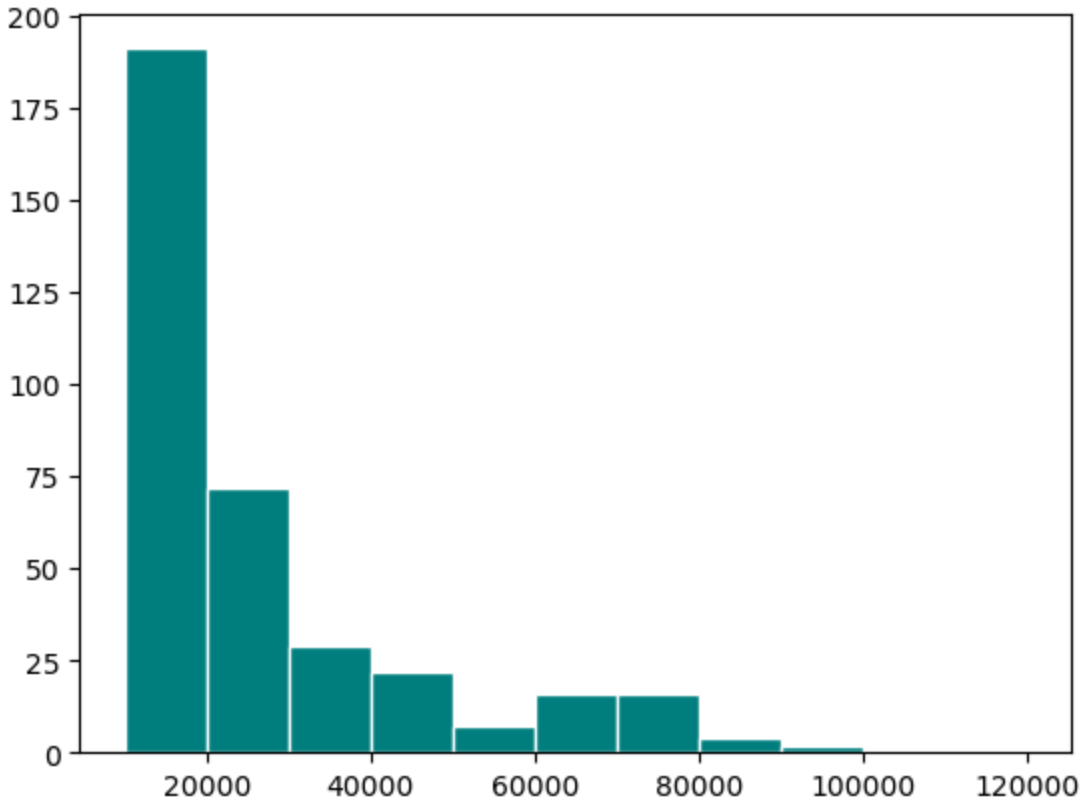
430 rows × 19 columns

In [ ]:
```

```

In [69]:
```
plt.hist(df.sales_price, bins=[10000, 20000, 30000, 40000, 50000, 60000, 7000
```

Out[69]:
```
(array([191.,  72.,  29.,  22.,   7.,  16.,  16.,   4.,   2.,   0.]),
 array([ 10000,  20000,  30000,  40000,  50000,  60000,  70000,  80000,
         90000, 100000, 120000]),
 <BarContainer object of 10 artists>)
```



Majority of customers buy mobile phones within a range of 10000 to 20000 which are budget-friendly smartphones

8. Which are the top 10 models by avg sales

In [78]:
```
avg_sales_by_model= df.groupby('model')['sales'].mean()
```

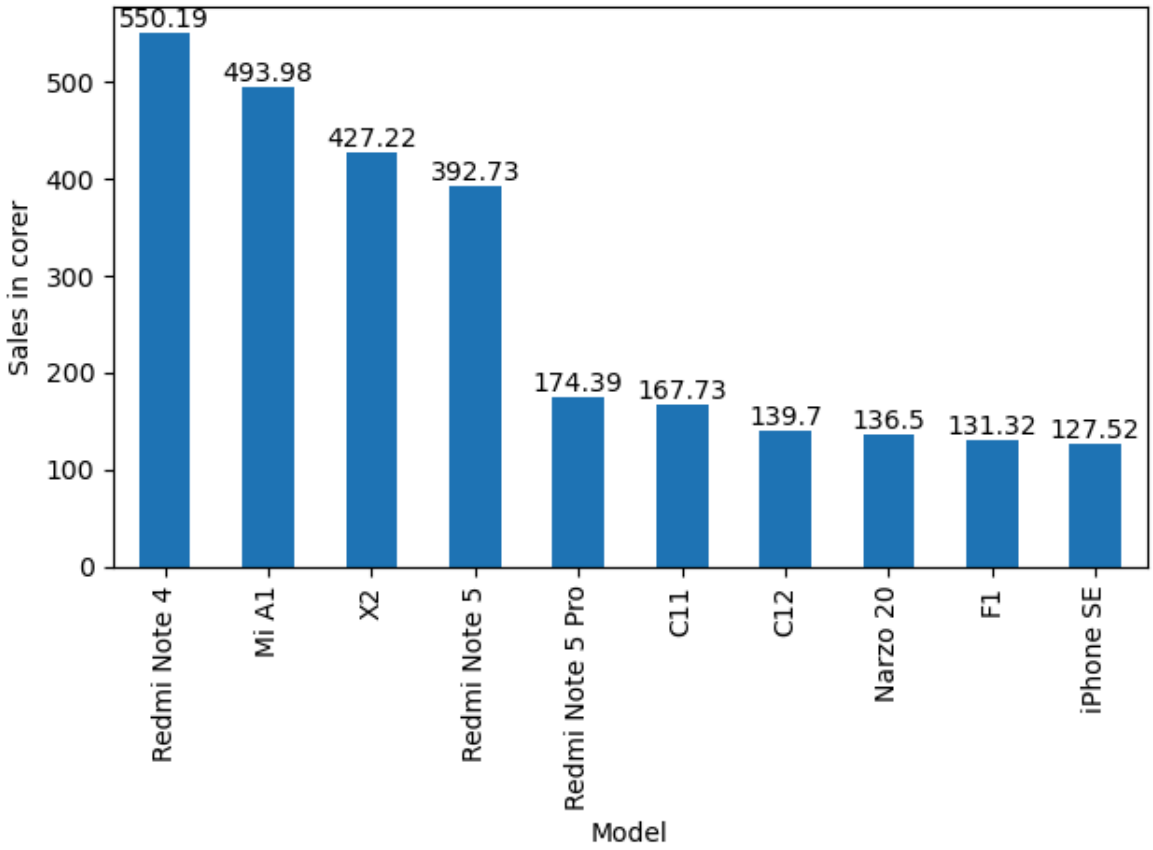
In [79]:
```
top_10_model= avg_sales_by_model.sort_values(ascending=False).head(10)
```

In [77]:
```
top_10_model
```

Out[77]:
```
model
Redmi Note 4        550.190
Mi A1               493.980
X2                  427.220
Redmi Note 5        392.730
Redmi Note 5 Pro    174.395
C11                 167.730
C12                 139.700
Narzo 20            136.495
F1                  131.320
iPhone SE           127.520
Name: sales, dtype: float64
```

In [81]:
```
top_10_model.plot(kind='bar')
plt.xlabel('Model')
plt.ylabel('Sales in corer')
for i, value in  enumerate(top_10_model):
    plt.text(i,value,str(round(value,2)), ha='center', va='bottom')

plt.tight_layout()
plt.show()
```



what is battery capacity by brand

In [86]:
```
branddf['battery_capacity'].value_counts()
```

Out[86]:

brand	battery_capacity	
Apple	2815	33
	2942	18
	1800	5
Poco	5000	26
	6000	15
	5065	6
	5160	6
	4000	2
	4500	1
Realme	5000	53
	4500	25
	6000	18
	4300	13
	4200	9
	4035	6
	4230	6
	4000	4
	4310	4
Samsung	5000	40
	6000	21
	4000	16
	4500	16
	3300	7
	7000	6
	4300	3
	4400	3
	2600	2
	3400	2
	3000	1
	3700	1
	4800	1
Xiaomi	4000	27
	5000	10
	4250	6
	5020	4
	4030	3
	4520	3
	3080	2
	4820	2
	3000	1
	3300	1
	4100	1
	4780	1

Name: battery_capacity, dtype: int64

10. How much sales created by each brand ?

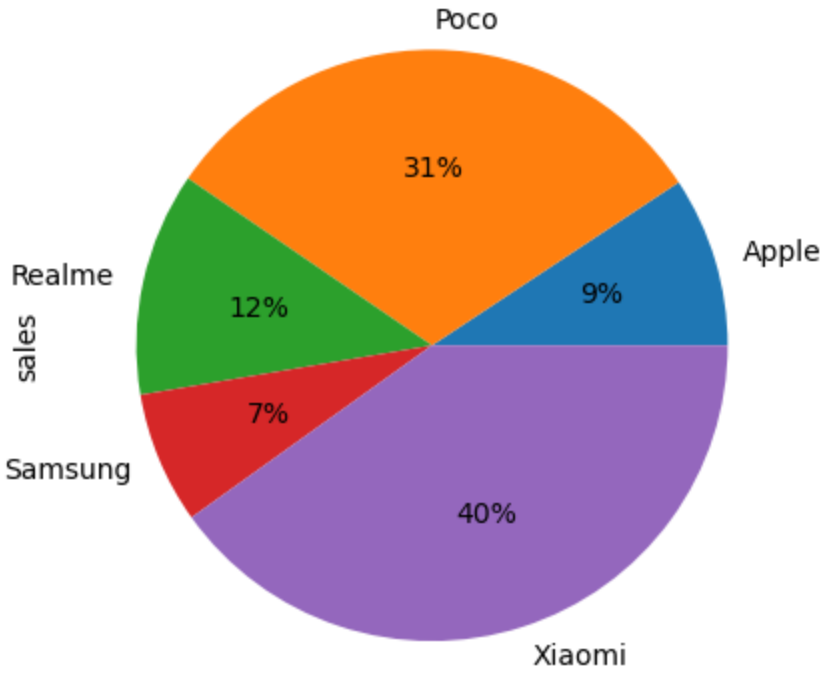
In [87]:
```
s=branddf.sales.max()
```

In [88]:
```
s
```

Out[88]:
```
brand
Apple      127.52
Poco       427.22
Realme     167.73
Samsung     98.89
Xiaomi     550.19
Name: sales, dtype: float64
```

In [89]:
```
s.plot(kind = "pie", autopct="%0.f%%")
```

Out[89]:
```
<AxesSubplot:ylabel='sales'>
```



Xiaomi generate 40% of maximum sales as compared to other brands