In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
import os
```

In [8]:
```
os.chdir("C:\\Users\\shyamu v\\Desktop")
```

In [9]:
```
data=pd.read_csv("test.csv")
data
```

Out[9]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2	0	1	0
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0	7	1	0	0
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10	10	0	1	1
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0	7	1	1	0
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8	7	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	1700	1	1.9	0	0	1	54	0.5	170	...	17	644	913	2121	14	8	15	1	1	0
996	997	609	0	1.8	1	0	0	13	0.9	186	...	2	1152	1632	1933	8	1	19	0	1	1
997	998	1185	0	1.4	0	1	1	8	0.5	80	...	12	477	825	1223	5	0	14	1	0	0
998	999	1533	1	0.5	1	0	0	50	0.4	171	...	12	38	832	2509	15	11	6	0	1	0
999	1000	1270	1	0.5	0	4	1	35	0.1	140	...	19	457	608	2828	9	2	3	1	0	1

1000 rows × 21 columns

In [14]:
```
data.head()
```

Out[14]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2	0	1	0
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0	7	1	0	0
2	3	1807	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10	10	0	1	1
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0	7	1	1	0
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8	7	1	0	1

5 rows × 21 columns

In [15]:
```
data.shape
```

Out[15]:
```
(1000, 21)
```

In [16]:
```
data.head(1)
```

Out[16]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2	0	1	0

1 rows × 21 columns

In [17]:
```
data.columns
```

Out[17]:
```
Index(['id', 'battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc',
       'four_g', 'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc',
       'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g',
       'touch_screen', 'wifi'],
      dtype='object')
```

In [18]:
```
data.index
```

Out[18]:
```
RangeIndex(start=0, stop=1000, step=1)
```

In [22]:
```
data.describe(include='all')
```

Out[22]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	...	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	1248.510000	0.516000	1.540900	0.517000	4.593000	0.487000	33.652000	0.517500	139.51100	...	10.054000	627.121000	1239.774000	2138.998000	11.995000	5.316000	11.085000	0.756000	0.500000	0.507000
std	288.819436	432.458227	0.499994	0.829268	0.499961	4.463325	0.500081	18.128694	0.280861	34.85155	...	6.095099	432.929699	439.670981	1088.092278	4.320607	4.240062	5.497636	0.429708	0.500025	0.500201
min	1.000000	500.000000	0.000000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.00000	...	0.000000	0.000000	501.000000	263.000000	5.000000	0.000000	2.000000	0.000000	0.000000	0.000000
25%	250.750000	895.000000	0.000000	0.700000	0.000000	1.000000	0.000000	18.000000	0.300000	109.75000	...	5.000000	283.750000	831.750000	1237.250000	8.000000	2.000000	6.750000	1.000000	0.000000	0.000000
50%	500.500000	1246.500000	1.000000	1.500000	1.000000	3.000000	0.000000	34.500000	0.500000	139.00000	...	10.000000	564.500000	1250.000000	2153.500000	12.000000	5.000000	11.000000	1.000000	0.500000	1.000000
75%	750.250000	1629.250000	1.000000	2.300000	1.000000	7.000000	1.000000	49.000000	0.800000	170.00000	...	16.000000	903.000000	1637.750000	3065.500000	16.000000	8.000000	16.000000	1.000000	1.000000	1.000000
max	1000.000000	1999.000000	1.000000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.00000	...	20.000000	1907.000000	1998.000000	3989.000000	19.000000	18.000000	20.000000	1.000000	1.000000	1.000000

8 rows × 21 columns

In [23]:
```
data['ram']
```

Out[23]:
```
0      3476
1      3895
2      2396
3      3893
4      1773
       ...
995    2121
996    1933
997    1223
998    2509
999    2828
Name: ram, Length: 1000, dtype: int64
```

In [24]:
```
data['battery_power'].max()
```

Out[24]:
```
1999
```

In [25]:
```
data['mobile_wt'].min()
```

Out[26]:
```
80
```

In [29]:
```
data.isnull().sum()
```

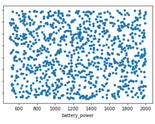
Out[29]:
```
id             0
battery_power  0
blue           0
clock_speed    0
dual_sim       0
fc             0
four_g         0
int_memory     0
m_dep          0
mobile_wt      0
n_cores        0
pc             0
px_height      0
px_width       0
ram            0
sc_h           0
sc_w           0
talk_time      0
three_g        0
touch_screen   0
wifi           0
dtype: int64
```

In [30]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             1000 non-null   int64  
 1   battery_power  1000 non-null   int64  
 2   blue           1000 non-null   int64  
 3   clock_speed    1000 non-null   float64
 4   dual_sim       1000 non-null   int64  
 5   fc             1000 non-null   int64  
 6   four_g         1000 non-null   int64  
 7   int_memory     1000 non-null   int64  
 8   m_dep          1000 non-null   float64
 9   mobile_wt      1000 non-null   int64  
 10  n_cores        1000 non-null   int64  
 11  pc             1000 non-null   int64  
 12  px_height      1000 non-null   int64  
 13  px_width       1000 non-null   int64  
 14  ram            1000 non-null   int64  
 15  sc_h           1000 non-null   int64  
 16  sc_w           1000 non-null   int64  
 17  talk_time      1000 non-null   int64  
 18  three_g        1000 non-null   int64  
 19  touch_screen   1000 non-null   int64  
 20  wifi           1000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 164.2 KB
```

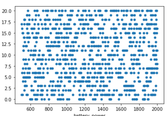
In [32]:
```
data.plot(x='battery_power',y='ram',kind='scatter')
```

Out[32]:
```
<AxesSubplot:xlabel="battery_power", ylabel="ram">
```



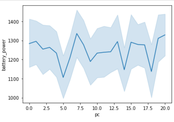
In [34]:
```
data.plot(x='battery_power',y='pc',kind='scatter')
```

Out[34]:
```
<AxesSubplot:xlabel="battery_power", ylabel="pc">
```

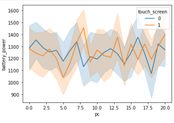


In [37]:
```
import seaborn as sns
```

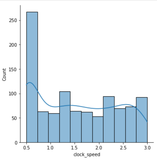
In [42]:
```
sns.lineplot(x="pc",y="battery_power",data=data)
plt.show()
```



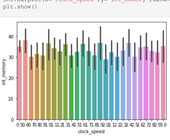
In [44]:
```
sns.lineplot(x="pc",y="battery_power",data=data,hue="touch_screen")
plt.show()
```



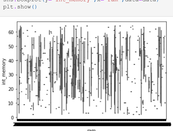
In [47]:
```
sns.displot(data["clock_speed"],kde=True)
plt.show()
```



In [51]:
```
sns.barplot(x="clock_speed",y="int_memory",data=data)
plt.show()
```



In [53]:
```
sns.boxplot(y="int_memory",x="ram",data=data)
plt.show()
```



In [ ]:
```

```