In [1]:

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

In [25]:

```python
fa= pd.read_csv('test.csv')
fa
```

Out[25]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mo
0	1	1043	1	1.8	1	14	0	5	0.1	
1	2	841	1	0.5	1	4	1	61	0.8	
2	3	1807	1	2.8	0	1	0	27	0.9	
3	4	1546	0	0.5	1	18	1	25	0.5	
4	5	1434	0	1.4	0	11	1	49	0.5	
...	...	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	0.5	
996	997	609	0	1.8	1	0	0	13	0.9	
997	998	1185	0	1.4	0	1	1	8	0.5	
998	999	1533	1	0.5	1	0	0	50	0.4	
999	1000	1270	1	0.5	0	4	1	35	0.1	

1000 rows × 21 columns

In [26]:

```python
fa.dropna()
```

Out[26]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mo
0	1	1043	1	1.8	1	14	0	5	0.1	
1	2	841	1	0.5	1	4	1	61	0.8	
2	3	1807	1	2.8	0	1	0	27	0.9	
3	4	1546	0	0.5	1	18	1	25	0.5	
4	5	1434	0	1.4	0	11	1	49	0.5	
...	...	...	...	...	...	...	...	...	...	
995	996	1700	1	1.9	0	0	1	54	0.5	
996	997	609	0	1.8	1	0	0	13	0.9	
997	998	1185	0	1.4	0	1	1	8	0.5	
998	999	1533	1	0.5	1	0	0	50	0.4	
999	1000	1270	1	0.5	0	4	1	35	0.1	

1000 rows × 21 columns

In [27]:
```
fa.drop_duplicates()
```

Out[27]:

ttery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc
1043	1	1.8	1	14	0	5	0.1	193	...	16
841	1	0.5	1	4	1	61	0.8	191	...	12
1807	1	2.8	0	1	0	27	0.9	186	...	4
1546	0	0.5	1	18	1	25	0.5	96	...	20
1434	0	1.4	0	11	1	49	0.5	108	...	18
...	...	...	...	...	...	...	...	...	...	...
1700	1	1.9	0	0	1	54	0.5	170	...	17
609	0	1.8	1	0	0	13	0.9	186	...	2
1185	0	1.4	0	1	1	8	0.5	80	...	12
1533	1	0.5	1	0	0	50	0.4	171	...	12
1270	1	0.5	0	4	1	35	0.1	140	...	19

1 columns

In [28]:
```
fa.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 [29]:
```
fa.describe()
```

Out[29]:

	id	battery_power	blue	clock_speed	dual_sim	fc
count	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
std	288.819436	432.458227	0.499994	0.829268	0.499961	4.463325
min	1.000000	500.000000	0.000000	0.500000	0.000000	0.000000
25%	250.750000	895.000000	0.000000	0.700000	0.000000	1.000000
50%	500.500000	1246.500000	1.000000	1.500000	1.000000	3.000000
75%	750.250000	1629.250000	1.000000	2.300000	1.000000	7.000000
max	1000.000000	1999.000000	1.000000	3.000000	1.000000	19.000000

8 rows × 21 columns

In [30]:
```
data=fa['battery_power'].value_counts()
data
```

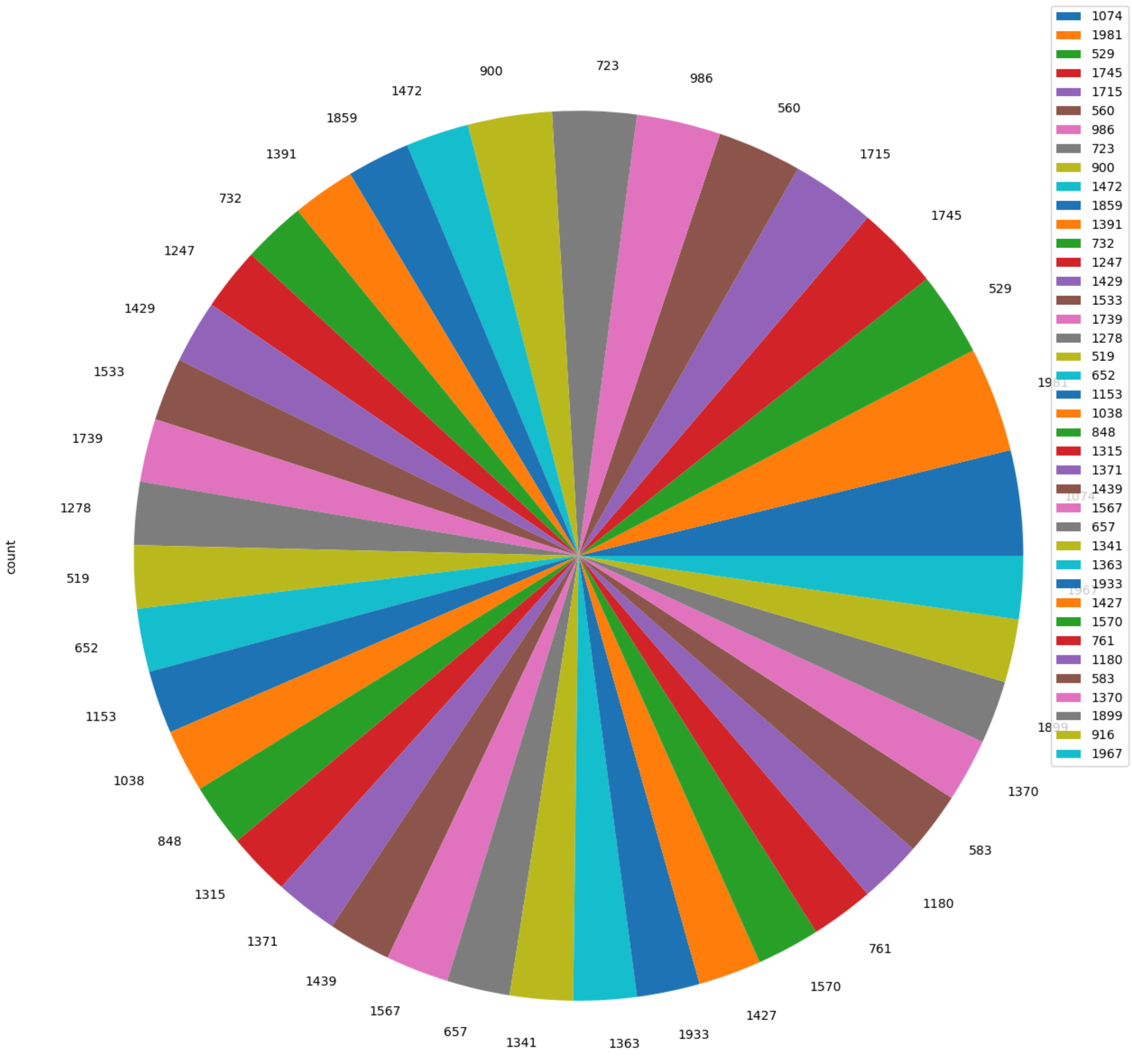
Out[30]:
```
battery_power
1074    5
1981    5
529     4
1745    4
1715    4
       ..
1248    1
1392    1
1706    1
1297    1
1185    1
Name: count, Length: 721, dtype: int64
```

In [31]:

```
newdata = data.head(n=40)
newdata.plot(kind='pie',figsize=(16,16))
plt.legend()
plt.title('pie chart ')
```

Out[31]:

```
<matplotlib.legend.Legend at 0x2189d0c23d0>
```



In [33]:

```
corelation = fa.corr()
corelation
```

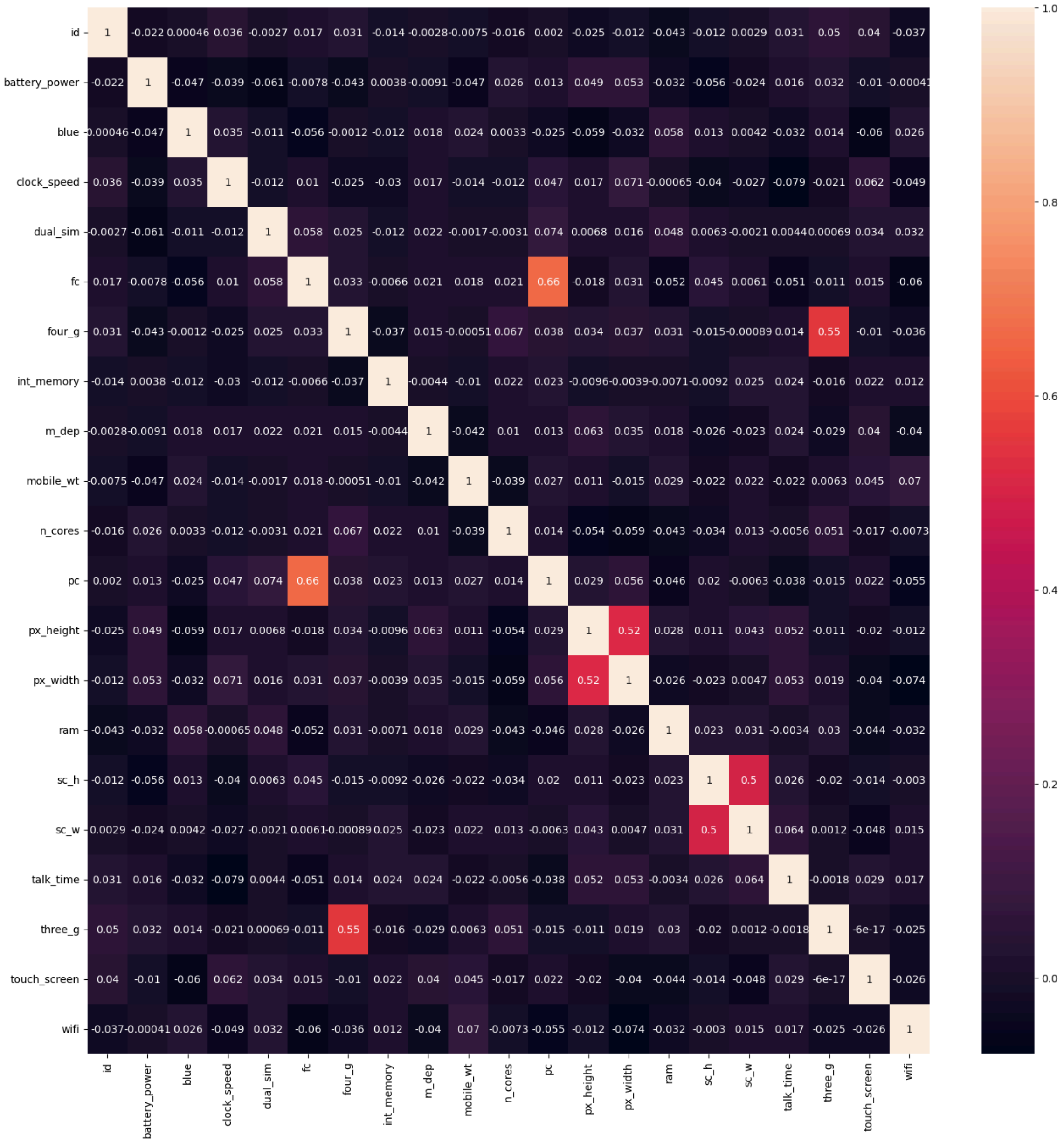
Out[33]:

	id	battery_power	blue	clock_speed	dual_sim	fc	fou
id	1.000000	-0.021511	0.000464	0.035917	-0.002721	0.016934	0.030
battery_power	-0.021511	1.000000	-0.046610	-0.039075	-0.061171	-0.007846	-0.042
blue	0.000464	-0.046610	1.000000	0.034754	-0.011100	-0.056063	-0.001
clock_speed	0.035917	-0.039075	0.034754	1.000000	-0.012423	0.010127	-0.024
dual_sim	-0.002721	-0.061171	-0.011100	-0.012423	1.000000	0.057606	0.024
fc	0.016934	-0.007846	-0.056063	0.010127	0.057606	1.000000	0.032
four_g	0.030921	-0.042520	-0.001169	-0.024665	0.024907	0.032832	1.000
int_memory	-0.014023	0.003751	-0.012416	-0.030487	-0.012158	-0.006565	-0.037
m_dep	-0.002794	-0.009065	0.018319	0.016995	0.021760	0.020859	0.014
mobile_wt	-0.007541	-0.047065	0.023513	-0.014107	-0.001734	0.018353	-0.000
n_cores	-0.015935	0.025732	0.003283	-0.012247	-0.003129	0.020828	0.066
pc	0.001969	0.012847	-0.025247	0.047469	0.073936	0.659338	0.037
px_height	-0.025056	0.048647	-0.058810	0.017277	0.006842	-0.017982	0.033
px_width	-0.012138	0.053365	-0.032054	0.070585	0.015610	0.030550	0.036
ram	-0.043442	-0.032366	0.057570	-0.000650	0.048171	-0.051997	0.030
sc_h	-0.011972	-0.055665	0.012780	-0.039503	0.006295	0.045158	-0.015
sc_w	0.002918	-0.023905	0.004223	-0.027138	-0.002064	0.006115	-0.000
talk_time	0.030807	0.015546	-0.031995	-0.078797	0.004390	-0.051458	0.013
three_g	0.049571	0.031514	0.013530	-0.021406	0.000690	-0.011121	0.553
touch_screen	0.039768	-0.010138	-0.060031	0.061893	0.034020	0.015467	-0.010
wifi	-0.036643	-0.000414	0.025568	-0.048593	0.031545	-0.060373	-0.035

21 rows × 21 columns

In [36]:
```
plt.figure(figsize=(18,18))
sns.heatmap(corelation,annot= True)
```

Out[36]:
```
<Axes: >
```



In [58]:
```
ndata=fa.set_index('id')
ndata
```

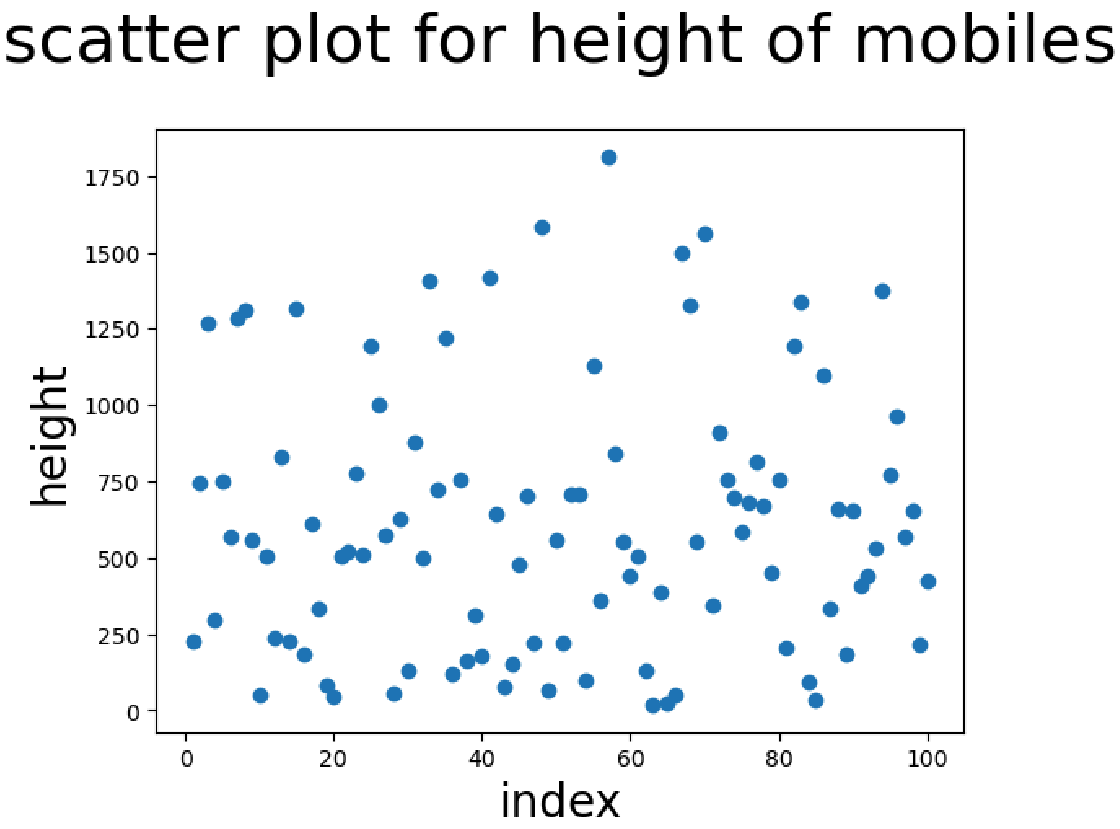
Out[58]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_v
id									
1	1043	1	1.8	1	14	0	5	0.1	19
2	841	1	0.5	1	4	1	61	0.8	19
3	1807	1	2.8	0	1	0	27	0.9	18
4	1546	0	0.5	1	18	1	25	0.5	9
5	1434	0	1.4	0	11	1	49	0.5	10
...	...	...	...	...	...	...	...	...	.
996	1700	1	1.9	0	0	1	54	0.5	17
997	609	0	1.8	1	0	0	13	0.9	18
998	1185	0	1.4	0	1	1	8	0.5	8
999	1533	1	0.5	1	0	0	50	0.4	17
1000	1270	1	0.5	0	4	1	35	0.1	14

1000 rows × 20 columns

In [57]:
```
sc = ndata.head(n=100)
x=sc.index
y=sc['px_height']
plt.scatter(x,y)
plt.title('scatter plot for height of mobiles',fontsize=30,pad=30)
plt.ylabel('height',fontsize=20)
plt.xlabel('index',fontsize=20)
```

Out[57]:
```
Text(0.5, 0, 'index')
```

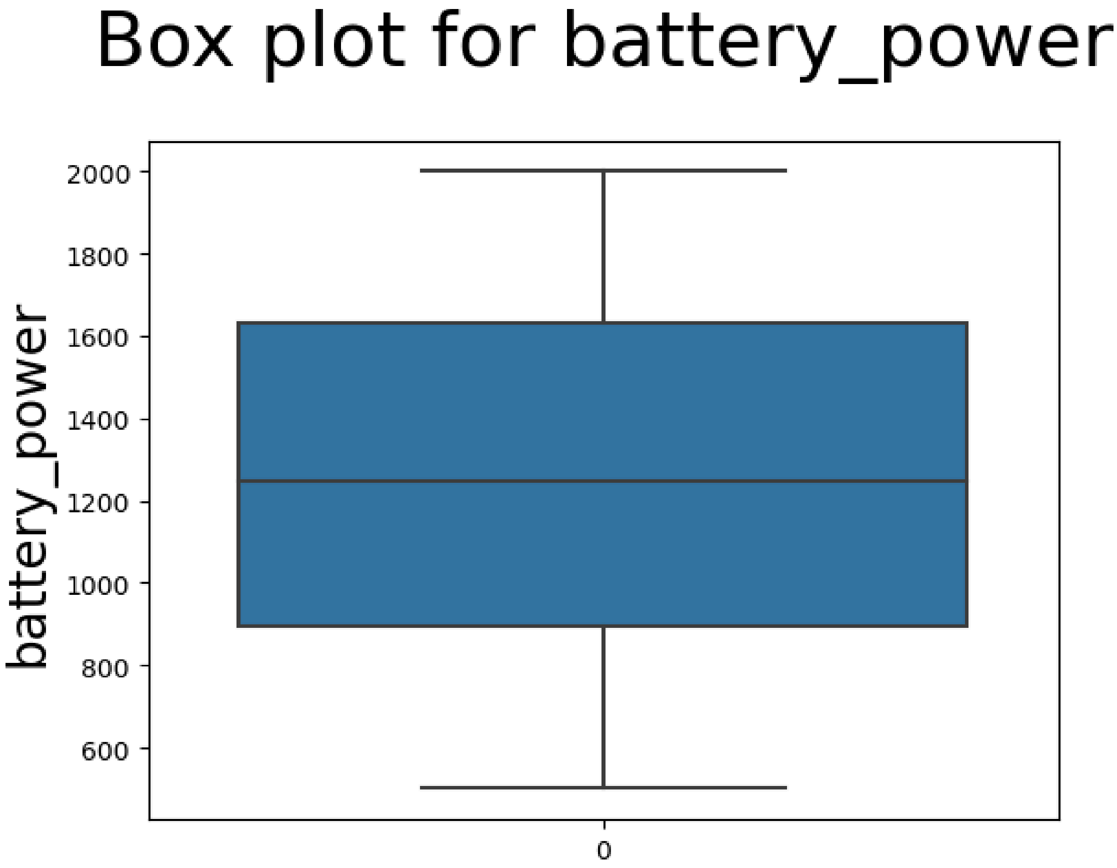


In [64]:

```
box=fa['battery_power']
sns.boxplot(box)
plt.title('Box plot for battery_power',fontsize= 30,pad=30)
plt.ylabel('battery_power',fontsize=20)
```

Out[64]:

```
Text(0, 0.5, 'battery_power')
```



In [ ]:

```
```