

```
In [15]: import pandas as pd
import numpy as np
from pandas import Series,DataFrame

import scipy
from scipy import stats
```

```
In [22]: df=pd.read_csv('C:/Users/SUBINDAS/Desktop/subin/mobile price classification.csv')
df.columns=['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', 'price_range']
```

```
In [5]: #showing first rows with head()
df.head()
```

```
Out[5]:
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	..
0	842	0	2.2	0	1	0	7	0.6	188	2	..
1	1021	1	0.5	1	0	1	53	0.7	136	3	..
2	563	1	0.5	1	2	1	41	0.9	145	5	..
3	615	1	2.5	0	0	0	10	0.8	131	6	..
4	1821	1	1.2	0	13	1	44	0.6	141	2	..

5 rows × 21 columns



```
In [21]: df.columns
```

```
Out[21]: Index(['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', 'price_range'],
      dtype='object')
```

```
In [25]: # max value per column
df.max()
```

```
Out[25]:
```

battery_power	1998.0
blue	1.0
clock_speed	3.0
dual_sim	1.0
fc	19.0
four_g	1.0
int_memory	64.0
m_dep	1.0
mobile_wt	200.0
n_cores	8.0
pc	20.0
px_height	1960.0
px_width	1998.0
ram	3998.0
sc_h	19.0
sc_w	18.0
talk_time	20.0
three_g	1.0

```
touch_screen      1.0
wifi              1.0
price_range       3.0
dtype: float64
```

```
In [26]: # max value per column
df.min()
```

```
Out[26]: battery_power    501.0
blue            0.0
clock_speed     0.5
dual_sim        0.0
fc              0.0
four_g          0.0
int_memory      2.0
m_dep           0.1
mobile_wt       80.0
n_cores         1.0
pc              0.0
px_height       0.0
px_width        500.0
ram             256.0
sc_h             5.0
sc_w             0.0
talk_time        2.0
three_g          0.0
touch_screen     0.0
wifi             0.0
price_range      0.0
dtype: float64
```

```
In [27]: # median
df.median()
```

```
Out[27]: battery_power    1226.0
blue            0.0
clock_speed     1.5
dual_sim        1.0
fc              3.0
four_g          1.0
int_memory      32.0
m_dep           0.5
mobile_wt       141.0
n_cores         4.0
pc              10.0
px_height       564.0
px_width        1247.0
ram             2146.5
sc_h             12.0
sc_w             5.0
talk_time        11.0
three_g          1.0
touch_screen     1.0
wifi             1.0
price_range      1.5
dtype: float64
```

```
In [28]: # standard deviation
df.std()
```

```
Out[28]: battery_power    439.418206
blue            0.500100
clock_speed     0.816004
dual_sim        0.500035
```

```
fc           4.341444
four_g        0.499662
int_memory   18.145715
m_dep         0.288416
mobile_wt     35.399655
n_cores       2.287837
pc            6.064315
px_height    443.780811
px_width     432.199447
ram          1084.732044
sc_h          4.213245
sc_w          4.356398
talk_time    5.463955
three_g       0.426273
touch_screen  0.500116
wifi          0.500076
price_range   1.118314
dtype: float64
```

```
In [29]: # variance
df.var()
```

```
Out[29]: battery_power 1.930884e+05
blue           2.501001e-01
clock_speed   6.658629e-01
dual_sim      2.500348e-01
fc             1.884813e+01
four_g         2.496626e-01
int_memory    3.292670e+02
m_dep          8.318353e-02
mobile_wt     1.253136e+03
n_cores        5.234197e+00
pc             3.677592e+01
px_height     1.969414e+05
px_width      1.867964e+05
ram            1.176644e+06
sc_h           1.775143e+01
sc_w           1.897820e+01
talk_time     2.985481e+01
three_g        1.817086e-01
touch_screen  2.501161e-01
wifi           2.500760e-01
price_range   1.250625e+00
dtype: float64
```

```
In [31]: # Lower quartile/first quartile
df.quantile(0.25)
```

```
Out[31]: battery_power    851.75
blue              0.00
clock_speed      0.70
dual_sim         0.00
fc               1.00
four_g           0.00
int_memory       16.00
m_dep            0.20
mobile_wt        109.00
n_cores          3.00
pc               5.00
px_height        282.75
px_width         874.75
ram              1207.50
sc_h             9.00
sc_w             2.00
```

```
talk_time      6.00
three_g       1.00
touch_screen   0.00
wifi          0.00
price_range    0.75
Name: 0.25, dtype: float64
```

```
In [32]: # second quartile/median
df.quantile(0.5)
```

```
Out[32]: battery_power    1226.0
blue           0.0
clock_speed    1.5
dual_sim       1.0
fc             3.0
four_g         1.0
int_memory     32.0
m_dep          0.5
mobile_wt      141.0
n_cores        4.0
pc             10.0
px_height      564.0
px_width       1247.0
ram            2146.5
sc_h           12.0
sc_w           5.0
talk_time      11.0
three_g        1.0
touch_screen   1.0
wifi           1.0
price_range    1.5
Name: 0.5, dtype: float64
```

```
In [33]: # upper quartile
df.quantile(0.75)
```

```
Out[33]: battery_power    1615.25
blue           1.00
clock_speed    2.20
dual_sim       1.00
fc             7.00
four_g         1.00
int_memory     48.00
m_dep          0.80
mobile_wt      170.00
n_cores        7.00
pc             15.00
px_height      947.25
px_width       1633.00
ram            3064.50
sc_h           16.00
sc_w           9.00
talk_time      16.00
three_g        1.00
touch_screen   1.00
wifi           1.00
price_range    2.25
Name: 0.75, dtype: float64
```

```
In [34]: # IQR
df.quantile(0.75)-df.quantile(0.25)
```

```
Out[34]: battery_power    763.50
blue           1.00
```

```

clock_speed      1.50
dual_sim         1.00
fc               6.00
four_g           1.00
int_memory       32.00
m_dep            0.60
mobile_wt        61.00
n_cores          4.00
pc               10.00
px_height        664.50
px_width         758.25
ram              1857.00
sc_h              7.00
sc_w              7.00
talk_time         10.00
three_g           0.00
touch_screen     1.00
wifi              1.00
price_range      1.50
dtype: float64

```

In [35]: `df.sum()`

```

Out[35]: battery_power    2477037.0
blue                 990.0
clock_speed         3044.5
dual_sim            1019.0
fc                  8619.0
four_g              1043.0
int_memory          64093.0
m_dep               1003.5
mobile_wt           280498.0
n_cores             9041.0
pc                  19833.0
px_height           1290216.0
px_width            2503031.0
ram                 4248426.0
sc_h                24613.0
sc_w                11534.0
talk_time           22022.0
three_g             1523.0
touch_screen        1006.0
wifi                1014.0
price_range         3000.0
dtype: float64

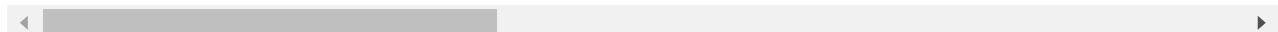
```

In [36]: `# generate descriptive statistics  
df.describe()`

	<b>battery_power</b>	<b>blue</b>	<b>clock_speed</b>	<b>dual_sim</b>	<b>fc</b>	<b>four_g</b>	<b>int_memory</b>	
<b>count</b>	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
<b>mean</b>	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	1
<b>std</b>	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	1
<b>min</b>	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	1
<b>25%</b>	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	1
<b>50%</b>	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	1
<b>75%</b>	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	1

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000

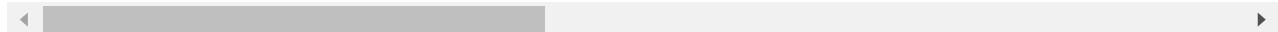
8 rows × 21 columns



In [37]: `# corelation  
df.corr()`

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	r
<b>battery_power</b>	1.000000	0.011252	0.011482	-0.041847	0.033334	0.015665	-0.004004	0.0
<b>blue</b>	0.011252	1.000000	0.021419	0.035198	0.003593	0.013443	0.041177	0.0
<b>clock_speed</b>	0.011482	0.021419	1.000000	-0.001315	-0.000434	-0.043073	0.006545	-0.0
<b>dual_sim</b>	-0.041847	0.035198	-0.001315	1.000000	-0.029123	0.003187	-0.015679	-0.0
<b>fc</b>	0.033334	0.003593	-0.000434	-0.029123	1.000000	-0.016560	-0.029133	-0.0
<b>four_g</b>	0.015665	0.013443	-0.043073	0.003187	-0.016560	1.000000	0.008690	-0.0
<b>int_memory</b>	-0.004004	0.041177	0.006545	-0.015679	-0.029133	0.008690	1.000000	0.0
<b>m_dep</b>	0.034085	0.004049	-0.014364	-0.022142	-0.001791	-0.001823	0.006886	1.0
<b>mobile_wt</b>	0.001844	-0.008605	0.012350	-0.008979	0.023618	-0.016537	-0.034214	0.0
<b>n_cores</b>	-0.029727	0.036161	-0.005724	-0.024658	-0.013356	-0.029706	-0.028310	-0.0
<b>pc</b>	0.031441	-0.009952	-0.005245	-0.017143	0.644595	-0.005598	-0.033273	0.0
<b>px_height</b>	0.014901	-0.006872	-0.014523	-0.020875	-0.009990	-0.019236	0.010441	0.0
<b>px_width</b>	-0.008402	-0.041533	-0.009476	0.014291	-0.005176	0.007448	-0.008335	0.0
<b>ram</b>	-0.000653	0.026351	0.003443	0.041072	0.015099	0.007313	0.032813	-0.0
<b>sc_h</b>	-0.029959	-0.002952	-0.029078	-0.011949	-0.011014	0.027166	0.037771	-0.0
<b>sc_w</b>	-0.021421	0.000613	-0.007378	-0.016666	-0.012373	0.037005	0.011731	-0.0
<b>talk_time</b>	0.052510	0.013934	-0.011432	-0.039404	-0.006829	-0.046628	-0.002790	0.0
<b>three_g</b>	0.011522	-0.030236	-0.046433	-0.014008	0.001793	0.584246	-0.009366	-0.0
<b>touch_screen</b>	-0.010516	0.010061	0.019756	-0.017117	-0.014828	0.016758	-0.026999	-0.0
<b>wifi</b>	-0.008343	-0.021863	-0.024471	0.022740	0.020085	-0.017620	0.006993	-0.0
<b>price_range</b>	0.200723	0.020573	-0.006606	0.017444	0.021998	0.014772	0.044435	0.0

21 rows × 21 columns



In [38]: `# covariance  
df.cov()`

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory
<b>battery_power</b>	193088.359838	2.472579	4.116922	-9.194773	63.592320	3.439322	-31.923572

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	...
<b>blue</b>	2.472579	0.250100	0.008741	0.008802	0.007801	0.003359	0.373669	C
<b>clock_speed</b>	4.116922	0.008741	0.665863	-0.000537	-0.001537	-0.017562	0.096914	-C
<b>dual_sim</b>	-9.194773	0.008802	-0.000537	0.250035	-0.063222	0.000796	-0.142263	-C
<b>fc</b>	63.592320	0.007801	-0.001537	-0.063222	18.848134	-0.035922	-2.295039	-C
<b>four_g</b>	3.439322	0.003359	-0.017562	0.000796	-0.035922	0.249663	0.078790	-C
<b>int_memory</b>	-31.923572	0.373669	0.096914	-0.142263	-2.295039	0.078790	329.266971	C
<b>m_dep</b>	4.319702	0.000584	-0.003381	-0.003193	-0.002243	-0.000263	0.036037	C
<b>mobile_wt</b>	28.689738	-0.152331	0.356738	-0.158945	3.629749	-0.292500	-21.977567	C
<b>n_cores</b>	-29.885322	0.041373	-0.010686	-0.028209	-0.132661	-0.033958	-1.175291	-C
<b>pc</b>	83.782186	-0.030183	-0.025955	-0.051983	16.970829	-0.016963	-3.661448	C
<b>px_height</b>	2905.736870	-1.525223	-5.259133	-4.632342	-19.247050	-4.265455	84.080518	E
<b>px_width</b>	-1595.644609	-8.977161	-3.341841	3.088397	-9.711403	1.608471	-65.366654	E
<b>ram</b>	-311.218050	14.294712	3.047585	22.277615	71.105629	3.963902	645.869530	-E
<b>sc_h</b>	-55.464653	-0.006221	-0.099970	-0.025174	-0.201462	0.057189	2.887692	-C
<b>sc_w</b>	-41.005692	0.001336	-0.026229	-0.036305	-0.234004	0.080550	0.927298	-C
<b>talk_time</b>	126.075334	0.038074	-0.050970	-0.107658	-0.161985	-0.127300	-0.276650	C
<b>three_g</b>	2.158241	-0.006446	-0.016151	-0.002986	0.003317	0.124440	-0.072446	-C
<b>touch_screen</b>	-2.310961	0.002516	0.008062	-0.004281	-0.032195	0.004188	-0.245012	-C
<b>wifi</b>	-1.833296	-0.005468	-0.009986	0.005686	0.043605	-0.004403	0.063456	-C
<b>price_range</b>	98.636568	0.011506	-0.006028	0.009755	0.106803	0.008254	0.901701	C

21 rows × 21 columns

In [39]: `import statistics as st`In [40]: `df.at[3:6,'batter_power']=20  
df`

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...
<b>0</b>	842	0	2.2	0	1	0	7	0.6	188	2	
<b>1</b>	1021	1	0.5	1	0	1	53	0.7	136	3	
<b>2</b>	563	1	0.5	1	2	1	41	0.9	145	5	
<b>3</b>	615	1	2.5	0	0	0	10	0.8	131	6	
<b>4</b>	1821	1	1.2	0	13	1	44	0.6	141	2	
...	...	...	...	...	...	...	...	...	...	...	..
<b>1995</b>	794	1	0.5	1	0	1	2	0.8	106	6	

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores
<b>1996</b>	1965	1	2.6	1	0	0	39	0.2	187	4
<b>1997</b>	1911	0	0.9	1	1	1	36	0.7	108	8
<b>1998</b>	1512	0	0.9	0	4	1	46	0.1	145	5
<b>1999</b>	510	1	2.0	1	5	1	45	0.9	168	6

2000 rows × 22 columns



```
In [41]: # average
st.mean(df['price_range'])
```

Out[41]: 1.5

```
In [42]: # harmonic_mean
st.harmonic_mean(df['price_range'])
```

Out[42]: 0

```
In [44]: # mode of data set
st.mode(df['clock_speed'])
```

Out[44]: 0.5

```
In [46]: # sample variance
st.variance(df['mobile_wt'])
```

Out[46]: 1253.1355667833916

```
In [47]: # population variance
st.pvariance(df['dual_sim'])
```

Out[47]: 0.24990975

```
In [48]: # sample deviation
st.stdev(df['ram'])
```

Out[48]: 1084.7320436099494

```
In [49]: # population deviation
st.pstdev(df['talk_time'])
```

Out[49]: 5.462589038175945

In [ ]: