

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

```
In [3]: # Будем использовать только обучающую выборку
df = pd.read_csv('test.csv')
df.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
```

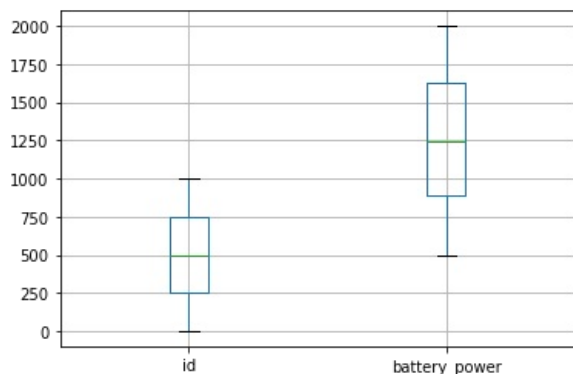
```
In [5]: col = pd.DataFrame(df, columns=["id","battery_power"])
col.head()
```

```
Out[5]:
```

	id	battery_power
0	1	1043
1	2	841
2	3	1807
3	4	1546
4	5	1434

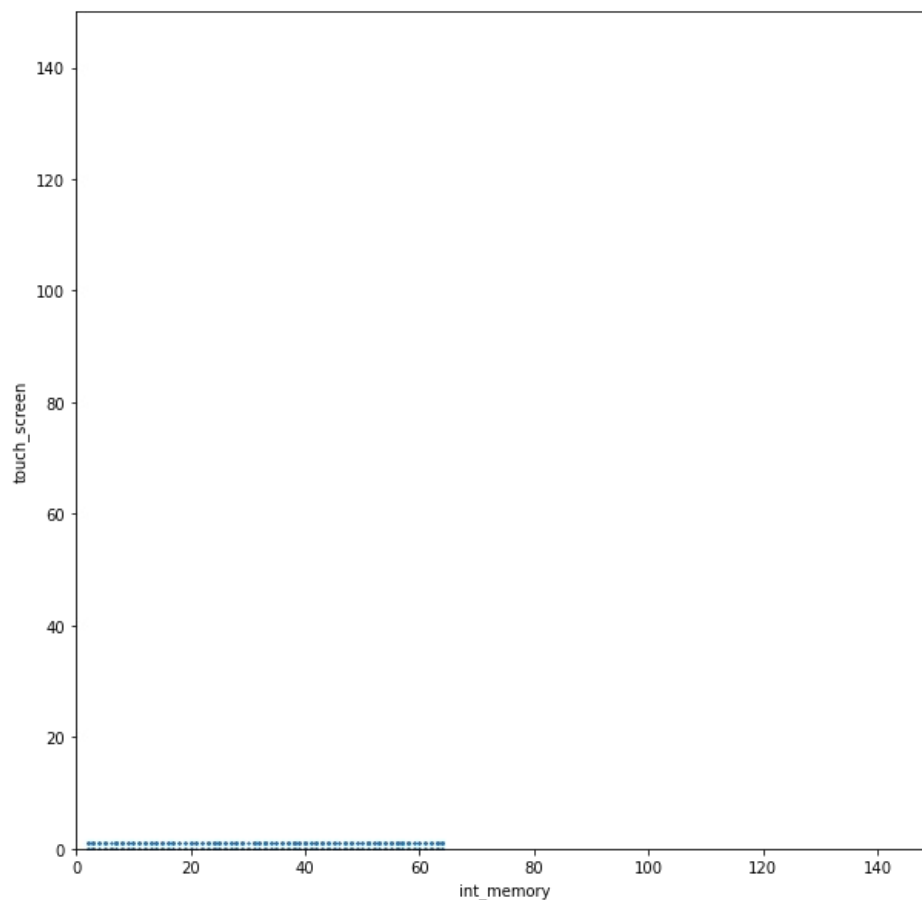
```
In [12]: #df['Age'].plot.hist(bins=10,grid=True)
df.boxplot(column=['id','battery_power'])
```

```
Out[12]: <AxesSubplot:>
```



```
In [16]: ax = df.plot.scatter(x='int_memory',y='touch_screen',s=1,figsize=(10,10))
ax.set_xlim(0, 150)
ax.set_ylim(0, 150)
```

```
Out[16]: (0.0, 150.0)
```



```
In [18]: hdata = df.loc[:,['id','battery_power']]
print('min(id) = ' + str(np.min(hdata['id'])))
print('max(id) = ' + str(np.max(hdata['id'])))
print('min(battery_power) = ' + str(np.min(hdata['battery_power'])))
print('max(battery_power) = ' + str(np.max(hdata['battery_power'])))
```

```
min(id) = 1
max(id) = 1000
min(battery_power) = 500
max(battery_power) = 1999
```

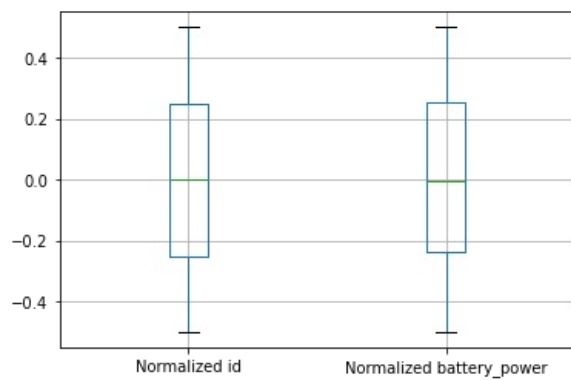
```
In [19]: hdata = hdata.apply(lambda x: (x - np.mean(x))/(np.max(x)-np.min(x)))
hdata.columns = ['Normalized id','Normalized battery_power']
hdata.head()
```

```
Out[19]:
```

	Normalized id	Normalized battery_power
0	-0.500000	-0.137098
1	-0.498999	-0.271855
2	-0.497998	0.372575
3	-0.496997	0.198459
4	-0.495996	0.123742

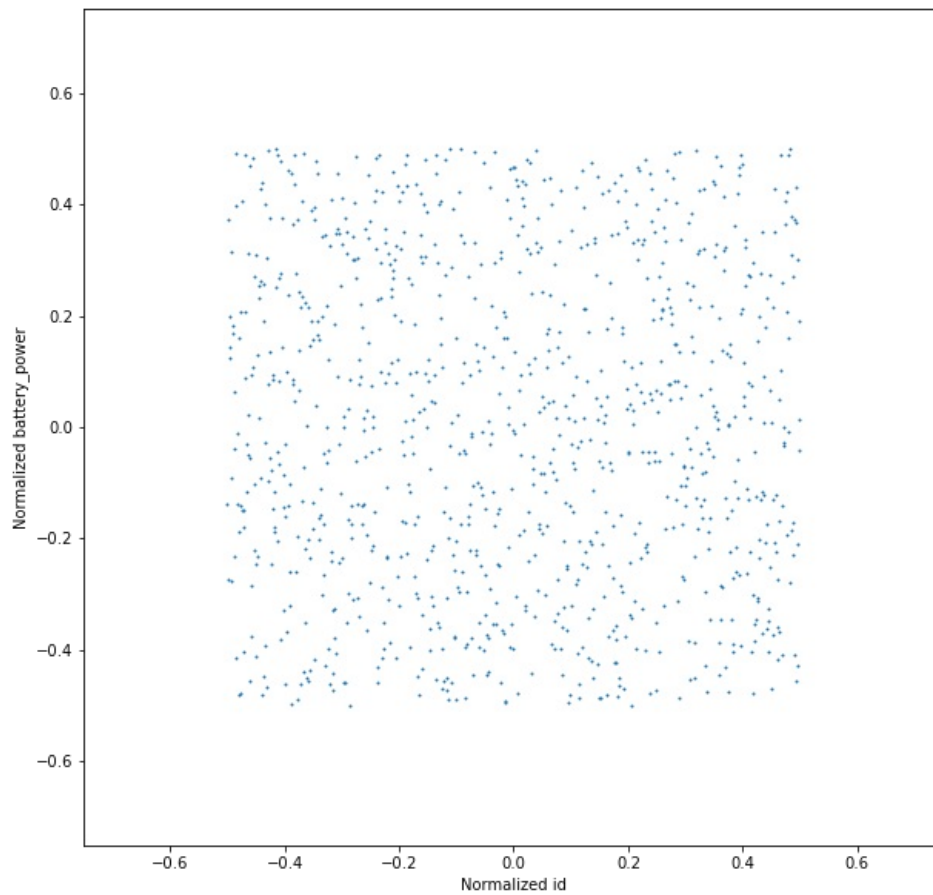
```
In [20]: hdata.boxplot(column=['Normalized id','Normalized battery_power'])
```

```
Out[20]: <AxesSubplot:>
```



```
In [21]: ax = hdata.plot.scatter(x='Normalized id',y='Normalized battery_power',s=1,figsize=(10,10))
ax.set_xlim(-0.75, 0.75)
ax.set_ylim(-0.75, 0.75)
```

```
Out[21]: (-0.75, 0.75)
```



```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js