jupyter Shasta_Tron (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

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

In [3]:
```
result = pd.read_csv("shastaTron.csv")
result.shape
```

Out[3]:
```
(1146, 9)
```

In [4]:
```
kk = result[["block_confirmation","Confirmed_time"]]
kk.head(10)
```

Out[4]:

	block_confirmation	Confirmed_time
0	Confirmed by 83 blocks	4 mins 43 secs ago
1	Confirmed by 54 blocks	3 mins 10 secs ago
2	Confirmed by 50 blocks	2 mins 56 secs ago
3	Confirmed by 64 blocks	3 mins 47 secs ago
4	Confirmed by 23 blocks	1 min 16 secs ago
5	Confirmed by 54 blocks	3 mins 9 secs ago
6	Confirmed by 86 blocks	5 mins 4 secs ago
7	Confirmed by 77 blocks	4 mins 27 secs ago
8	Confirmed by 46 blocks	2 mins 45 secs ago
9	Confirmed by 38 blocks	2 mins 4 secs ago

In [5]:
```
def block(k):
    if k!=k:
        return 0
    return int(k[13:15])

def timec(k):
    r=0
    i=0
    j=""
    while k[i]!=" ":
        j+=k[i]
        i+=1
    if k[i+1]=="m":
        r=int(j)*60
    elif k[i+1]=="s":
        return int(j)
    i+=1
    while k[i]!=" ":
        i+=1
    if k[i+1]=="a":
        return r
    j=""
    i+=1
    while k[i]!=" ":
        j+=k[i]
        i+=1
    return r+int(j)
```

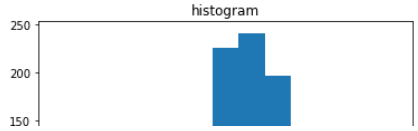
In [6]:
```
arr = []
for i in range(0,result.shape[0]):
    d = block(result['block_confirmation'].iloc[i])
    if d!=0:
        t=timec((result['Confirmed_time'].iloc[i]))
        arr.append(7*t/d)
```

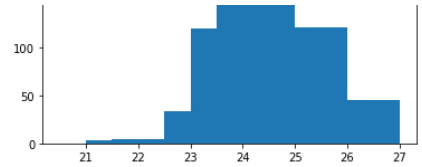
In [7]:
```
df = pd.DataFrame(arr)
df.describe()
```

Out[7]:

	0
count	1018.000000
mean	26.305858
std	21.165783
min	21.000000
25%	23.722222
50%	24.220307
75%	24.782282
max	256.200000

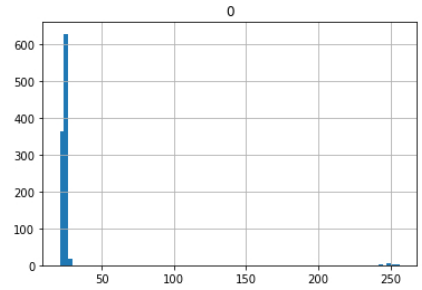
In [10]:
```
plt.hist(arr, bins = [20.5,21,21.5,22,22.5,23,23.5,24,24.5,25,26,27])
plt.title("histogram")
plt.show()
```





In [11]:
```
df.hist(bins=80)
```

Out[11]:
```
array([[<AxesSubplot:title={'center':'0'}>]], dtype=object)
```



In [12]:
```
df.describe()
```

Out[12]:

	0
count	1018.000000
mean	26.305858
std	21.165783
min	21.000000
25%	23.722222
50%	24.220307
75%	24.782282
max	256.200000

In [13]:
```
df.mean()
```

Out[13]:
```
0    26.305858
dtype: float64
```

In [ ]:
```

```