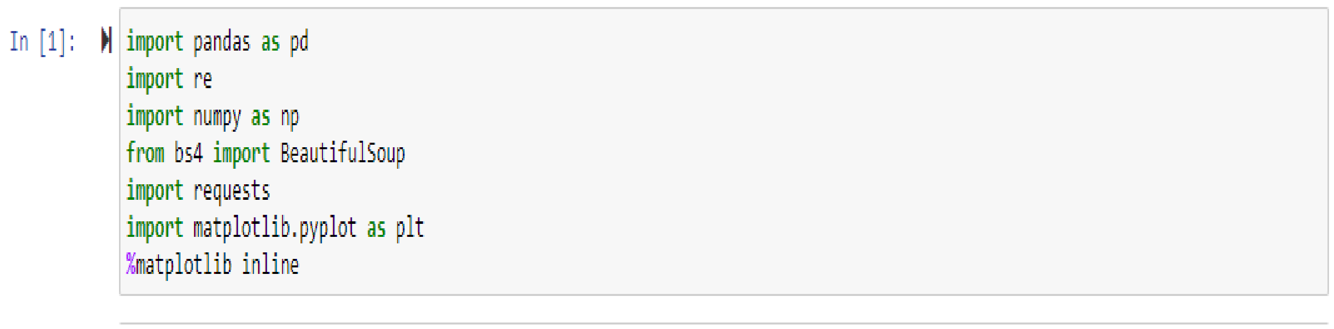
Screen Scraping

Ubaid Ullah Tariq

AIT - 580

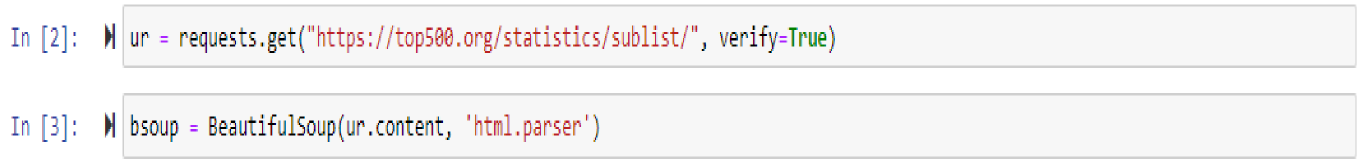
Importing essential libraries:



In [1]:

```
import pandas as pd
import re
import numpy as np
from bs4 import BeautifulSoup
import requests
import matplotlib.pyplot as plt
%matplotlib inline
```

Requesting to extract data from website:



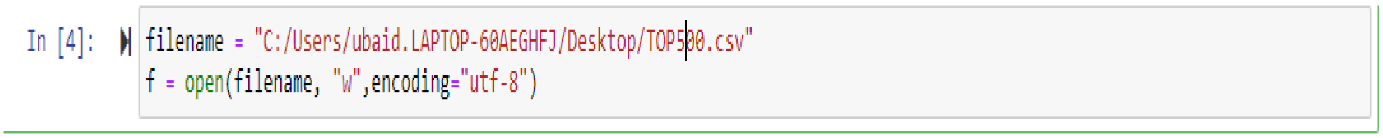
In [2]:

```
ur = requests.get("https://top500.org/statistics/sublist/", verify=True)
```

In [3]:

```
bsoup = BeautifulSoup(ur.content, 'html.parser')
```

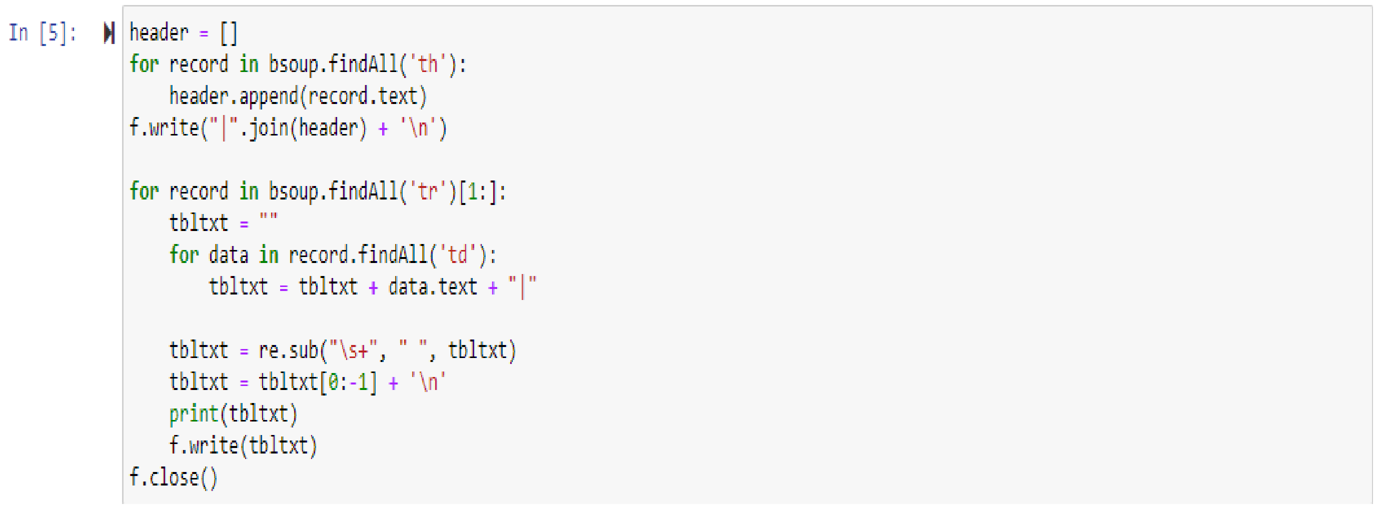
Loading CSV file:



In [4]:

```
filename = "C:/Users/ubaid.LAPTOP-60AEGHFJ/Desktop/TOP500.csv"
f = open(filename, "w",encoding="utf-8")
```

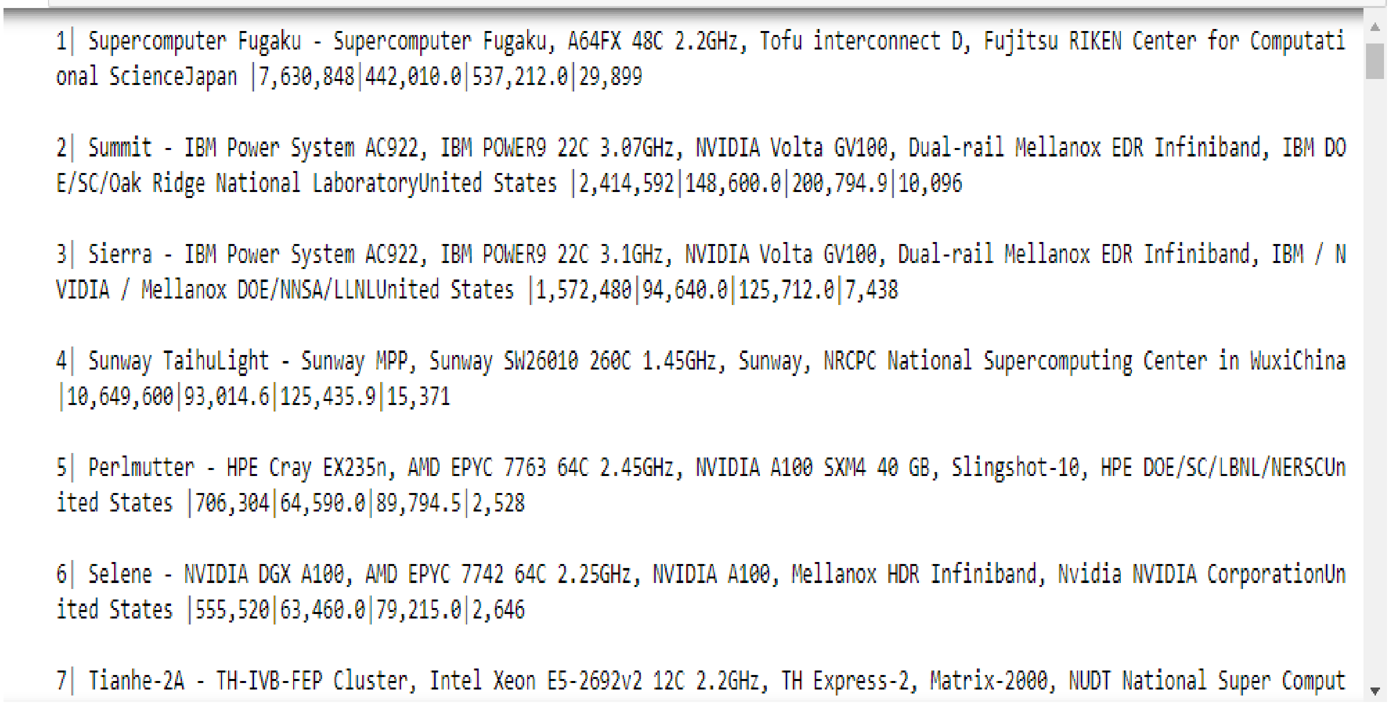
Header Code:



In [5]:

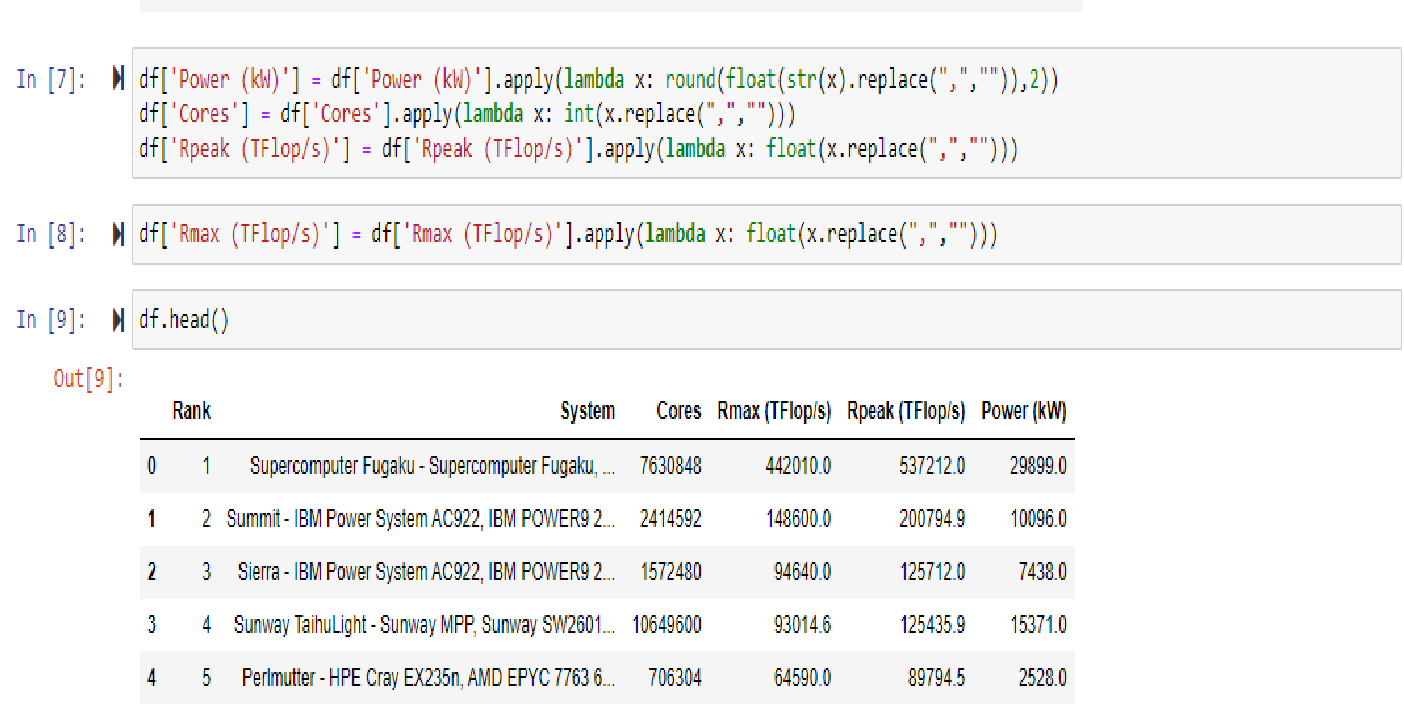
```
header = []
for record in bsoup.findAll('th'):
    header.append(record.text)
f.write("|".join(header) + '\n')

for record in bsoup.findAll('tr')[1:]:
    tbltxt = ""
    for data in record.findAll('td'):
        tbltxt = tbltxt + data.text + "|"

    tbltxt = re.sub("\s+", " ", tbltxt)
    tbltxt = tbltxt[0:-1] + '\n'
    print(tbltxt)
    f.write(tbltxt)
f.close()
```

Output:



```
1| Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computati
onal ScienceJapan |7,630,848|442,010.0|537,212.0|29,899

2| Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DO
E/SC/Oak Ridge National LaboratoryUnited States |2,414,592|148,600.0|200,794.9|10,096

3| Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / N
VIDIA / Mellanox DOE/NNSA/LLNLUnited States |1,572,480|94,640.0|125,712.0|7,438

4| Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in WuxiChina
|10,649,600|93,014.6|125,435.9|15,371

5| Perlmutter - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSCUn
ited States |706,304|64,590.0|89,794.5|2,528

6| Selene - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA CorporationUn
ited States |555,520|63,460.0|79,215.0|2,646

7| Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Comput
```

Summary Stats:



In [7]:
```
df['Power (kW)'] = df['Power (kW)'].apply(lambda x: round(float(str(x).replace(",","")),2))
df['Cores'] = df['Cores'].apply(lambda x: int(x.replace(",","")))
df['Rpeak (TFlop/s)'] = df['Rpeak (TFlop/s)'].apply(lambda x: float(x.replace(",","")))
```

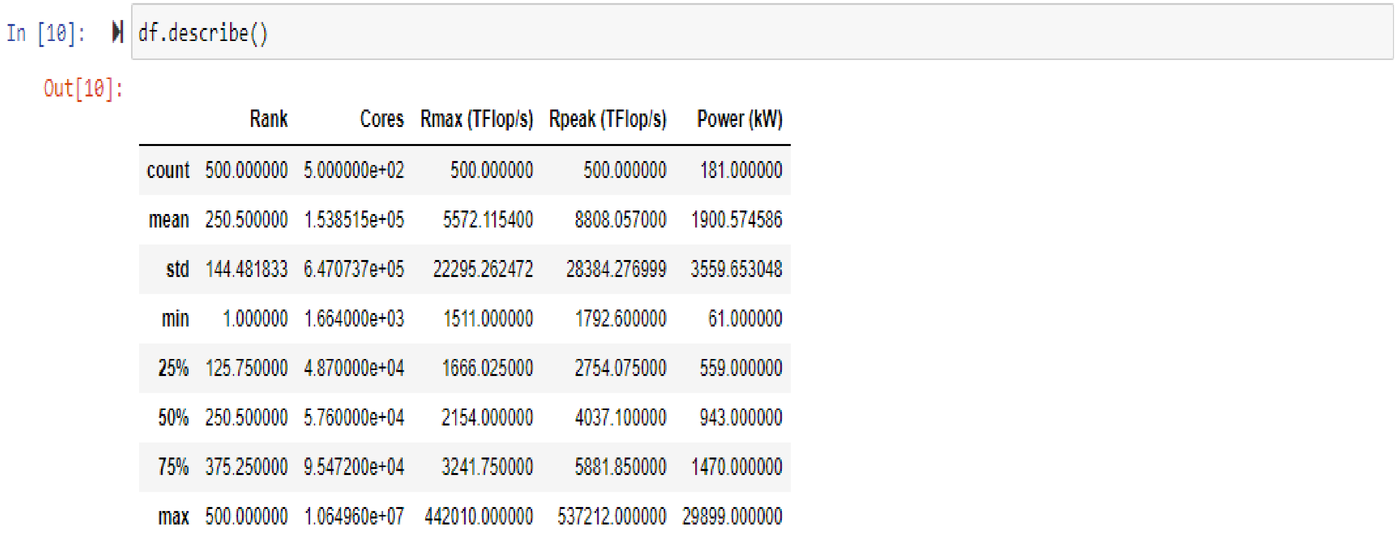
In [8]:
```
df['Rmax (TFlop/s)'] = df['Rmax (TFlop/s)'].apply(lambda x: float(x.replace(",","")))
```

In [9]:
```
df.head()
```

Out[9]:

	Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
0	1	Supercomputer Fugaku - Supercomputer Fugaku, ...	7630848	442010.0	537212.0	29899.0
1	2	Summit - IBM Power System AC922, IBM POWER9 2...	2414592	148600.0	200794.9	10096.0
2	3	Sierra - IBM Power System AC922, IBM POWER9 2...	1572480	94640.0	125712.0	7438.0
3	4	Sunway TaihuLight - Sunway MPP, Sunway SW2601...	10649600	93014.6	125435.9	15371.0
4	5	Perlmutter - HPE Cray EX235n, AMD EPYC 7763 6...	706304	64590.0	89794.5	2528.0

Summary Stats Output:



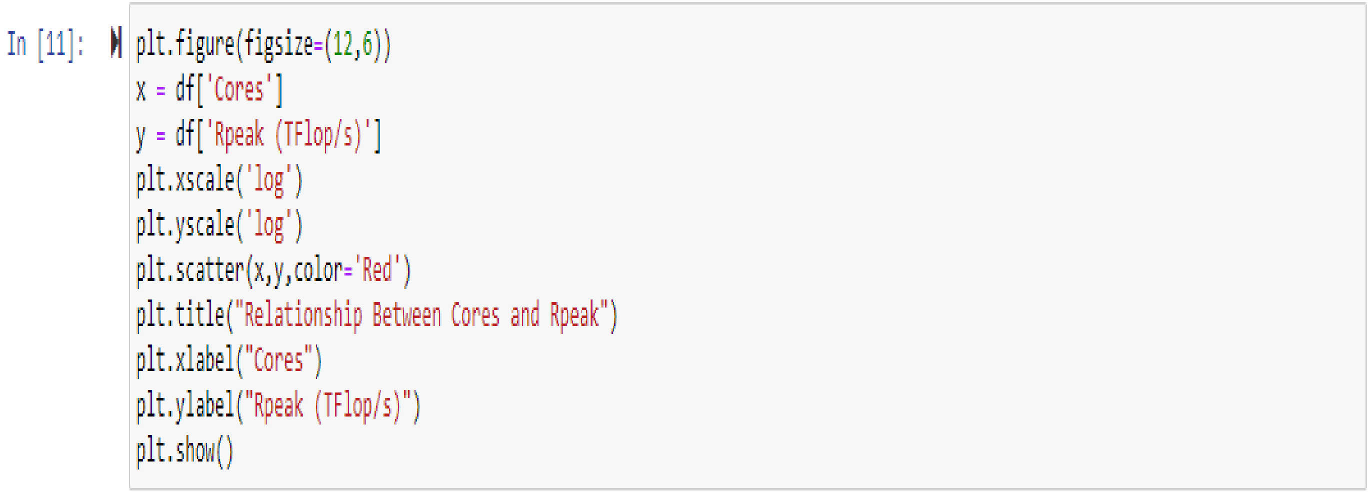
In [10]:

```
df.describe()
```

Out[10]:

	Rank	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
count	500.000000	5.000000e+02	500.000000	500.000000	181.000000
mean	250.500000	1.538515e+05	5572.115400	8808.057000	1900.574586
std	144.481833	6.470737e+05	22295.262472	28384.276999	3559.653048
min	1.000000	1.664000e+03	1511.000000	1792.600000	61.000000
25%	125.750000	4.870000e+04	1666.025000	2754.075000	559.000000
50%	250.500000	5.760000e+04	2154.000000	4037.100000	943.000000
75%	375.250000	9.547200e+04	3241.750000	5881.850000	1470.000000
max	500.000000	1.064960e+07	442010.000000	537212.000000	29899.000000

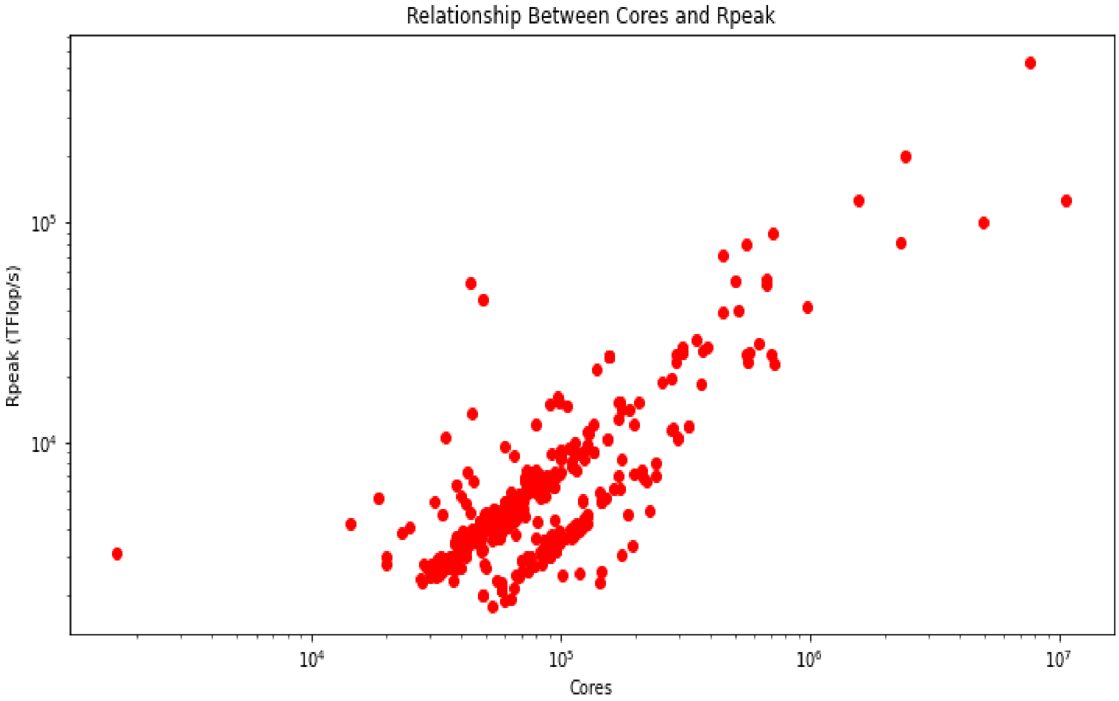
Visualization of Relationship Between Cores and Rpeak:



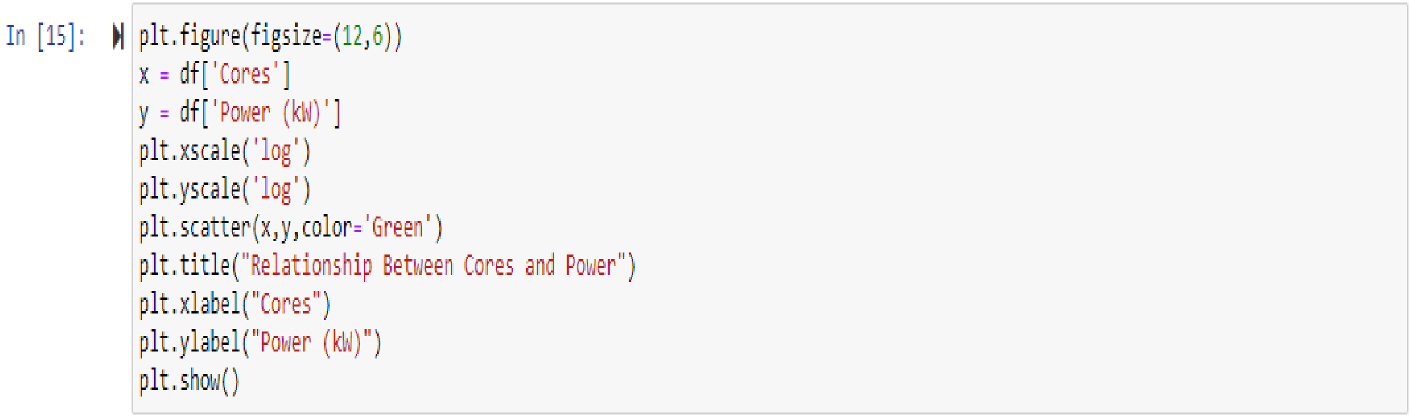
In [11]:

```
plt.figure(figsize=(12,6))
x = df['Cores']
y = df['Rpeak (TFlop/s)']
plt.xscale('log')
plt.yscale('log')
plt.scatter(x,y,color='Red')
plt.title("Relationship Between Cores and Rpeak")
plt.xlabel("Cores")
plt.ylabel("Rpeak (TFlop/s)")
plt.show()
```

Output:

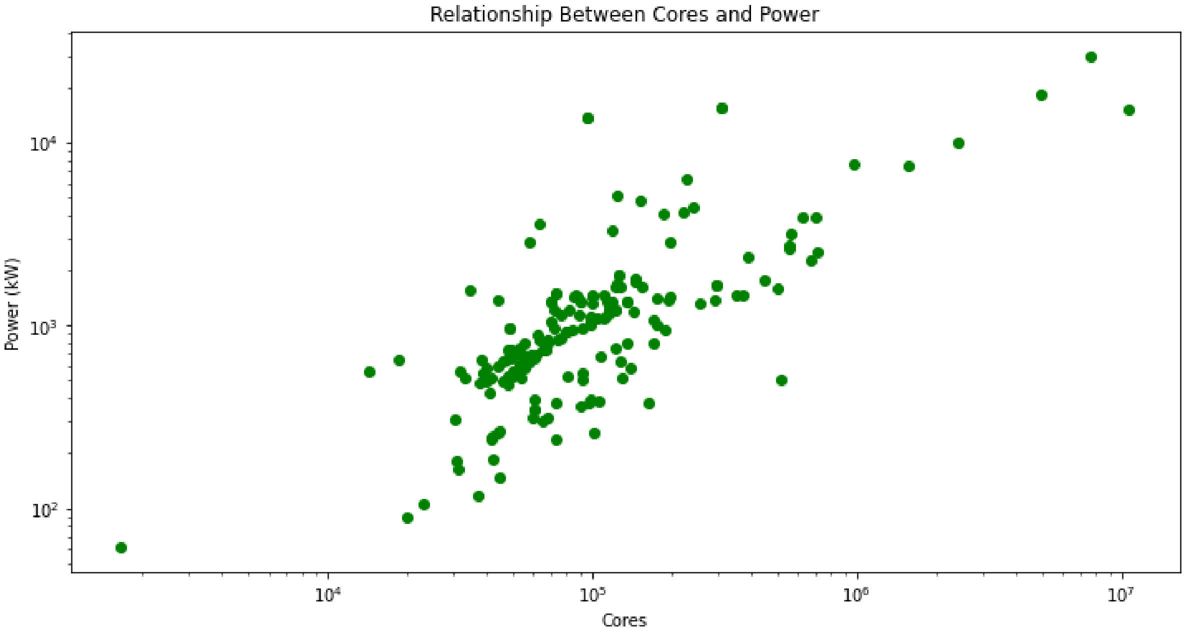


Visualization of Relationship Between Cores and Power:

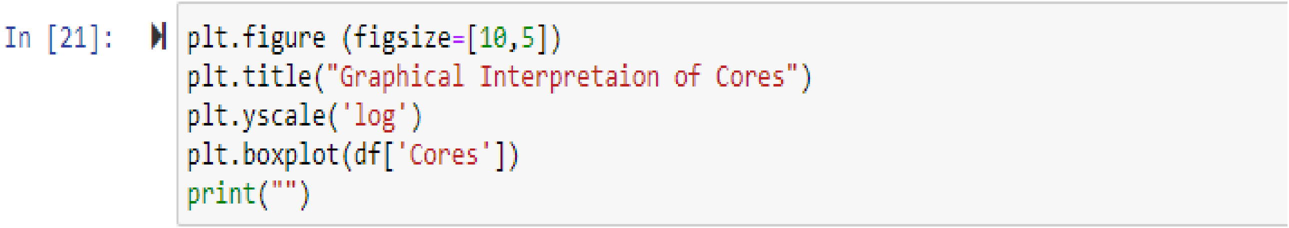

In [15]:

```
plt.figure(figsize=(12,6))
x = df['Cores']
y = df['Power (kW)']
plt.xscale('log')
plt.yscale('log')
plt.scatter(x,y,color='Green')
plt.title("Relationship Between Cores and Power")
plt.xlabel("Cores")
plt.ylabel("Power (kW)")
plt.show()
```

Output:

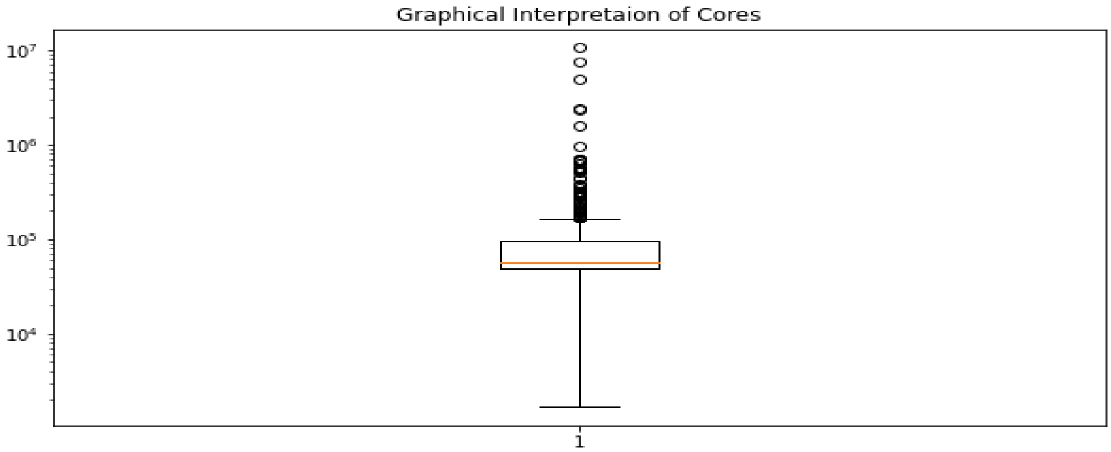


Graphical Interpretation of Cores:

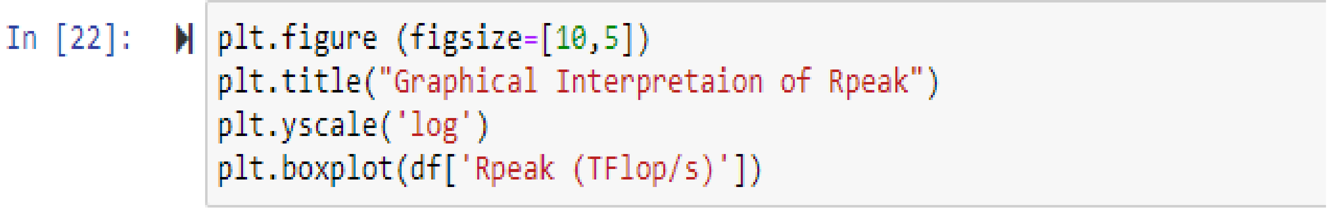


```
In [21]:  plt.figure (figsize=[10,5])
          plt.title("Graphical Interpretaion of Cores")
          plt.yscale('log')
          plt.boxplot(df['Cores'])
          print("")
```

Output:

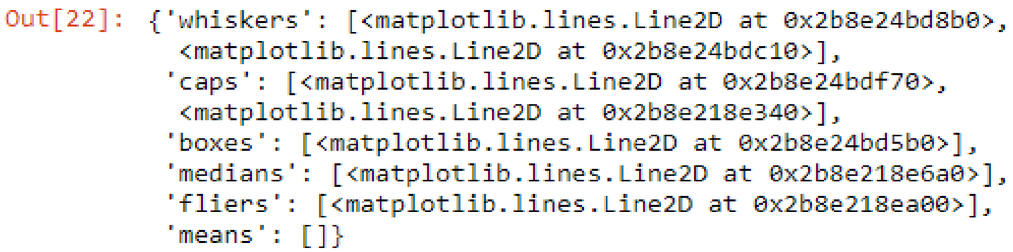


Graphical Interpretation of Rpeak:

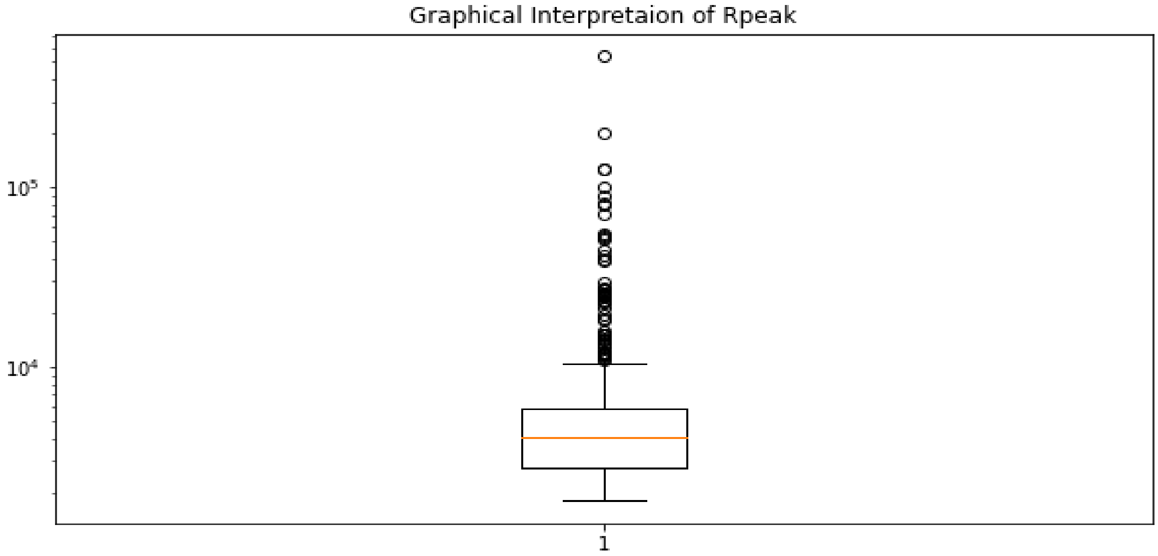


```
In [22]:  plt.figure (figsize=[10,5])
          plt.title("Graphical Interpretaion of Rpeak")
          plt.yscale('log')
          plt.boxplot(df['Rpeak (TFlop/s)'])
```

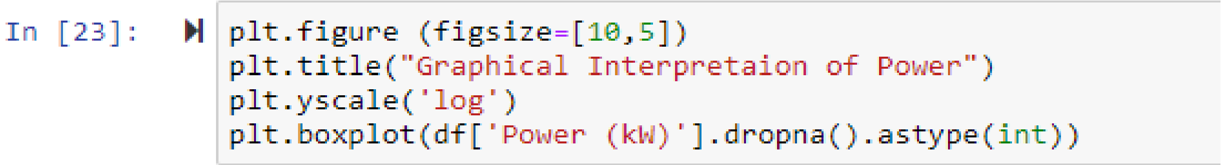
Output:



```
Out[22]: {'whiskers': [<matplotlib.lines.Line2D at 0x2b8e24bd8b0>,
           <matplotlib.lines.Line2D at 0x2b8e24bdc10>],
          'caps': [<matplotlib.lines.Line2D at 0x2b8e24bdf70>,
           <matplotlib.lines.Line2D at 0x2b8e218e340>],
          'boxes': [<matplotlib.lines.Line2D at 0x2b8e24bd5b0>],
          'medians': [<matplotlib.lines.Line2D at 0x2b8e218e6a0>],
          'fliers': [<matplotlib.lines.Line2D at 0x2b8e218ea00>],
          'means': []}
```

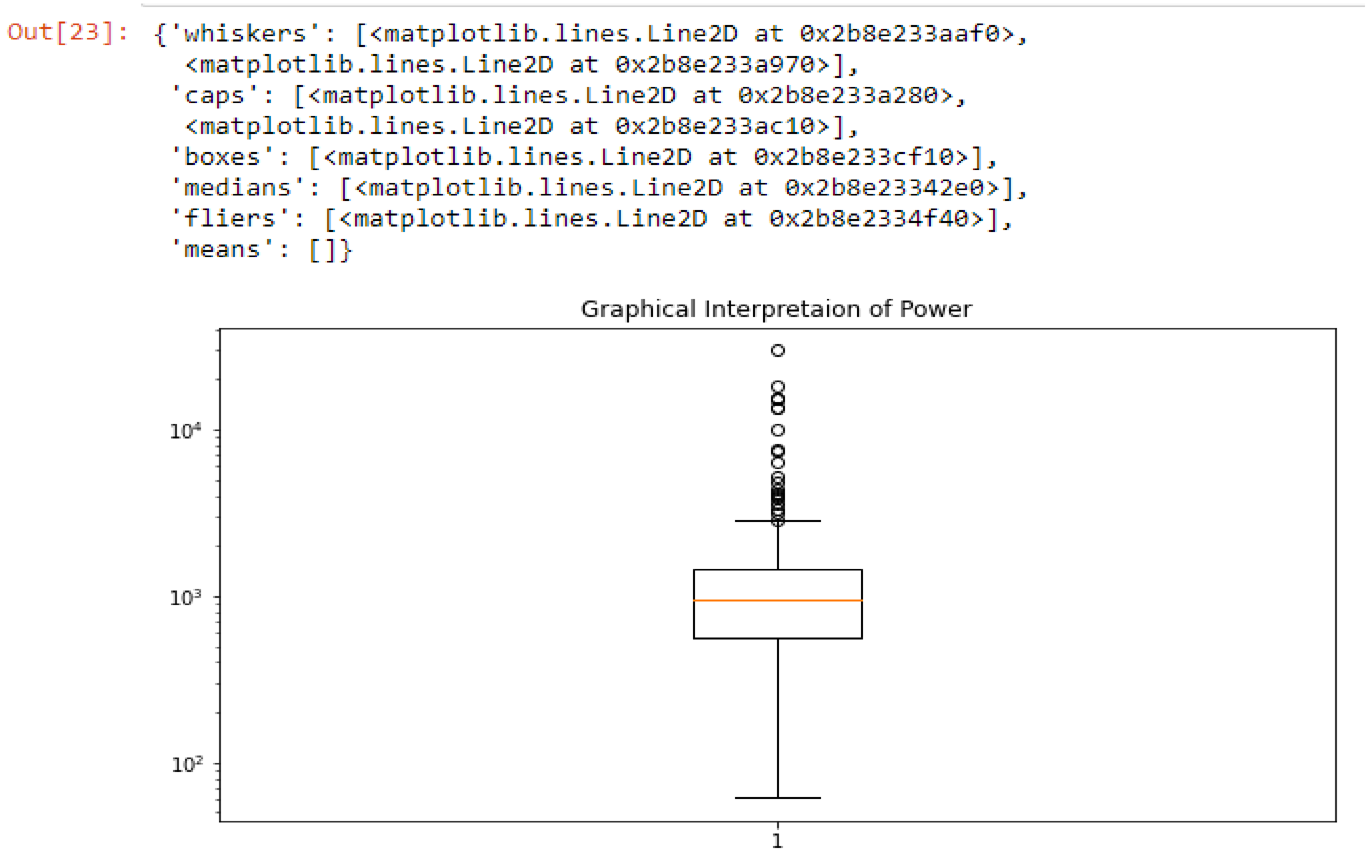


Graphical Interpretation of Power:



```
In [23]:  plt.figure (figsize=[10,5])
          plt.title("Graphical Interpretaion of Power")
          plt.yscale('log')
          plt.boxplot(df['Power (kW)'].dropna().astype(int))
```

Output:



```
Out[23]: {'whiskers': [<matplotlib.lines.Line2D at 0x2b8e233aaf0>,
           <matplotlib.lines.Line2D at 0x2b8e233a970>],
          'caps': [<matplotlib.lines.Line2D at 0x2b8e233a280>,
           <matplotlib.lines.Line2D at 0x2b8e233ac10>],
          'boxes': [<matplotlib.lines.Line2D at 0x2b8e233cf10>],
          'medians': [<matplotlib.lines.Line2D at 0x2b8e23342e0>],
          'fliers': [<matplotlib.lines.Line2D at 0x2b8e2334f40>],
          'means': []}
```