Baizhong Hou
Prof. Harry Foxwell
AIT-580-P02
Oct, 13th, 2019

1.

a)
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

pd.set_option('display.max_columns', None)
df=pd.read_csv("Atlantic.csv")
print("summary statistics:")
print(df.describe())

sns.countplot(x="year", data=df)
plt.show()

sns.countplot(x="status_of_system", data=df)
plt.show()

sns.distplot(df["max_sustained_wind"])
plt.show()
```

In [22]:
```
df=pd.read_csv("Atlantic.csv")
```

In [24]:
```
df.head()
```

Out[24]:

	basin	name	year	cyclone_of_the_year	date	time	status_of_system	latitude	longitude	max_sustained_wind	central_pressure
0	AL	UNNAMED	1851	1	18510625	0	HU	28.0N	94.8W	80	NaN
1	AL	UNNAMED	1851	1	18510625	600	HU	28.0N	95.4W	80	NaN
2	AL	UNNAMED	1851	1	18510625	1200	HU	28.0N	96.0W	80	NaN
3	AL	UNNAMED	1851	1	18510625	1800	HU	28.1N	96.5W	80	NaN
4	AL	UNNAMED	1851	1	18510625	2100	HU	28.2N	96.8W	80	NaN

In [25]:
```
df.info()
```

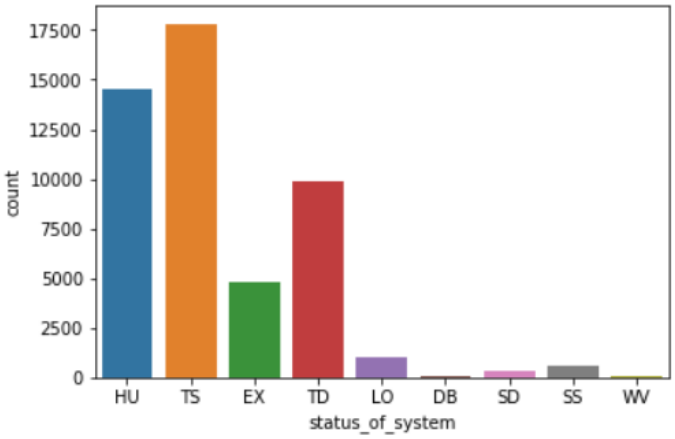
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49105 entries, 0 to 49104
Data columns (total 11 columns):
basin                49105 non-null object
name                 49105 non-null object
year                 49105 non-null int64
cyclone_of_the_year  49105 non-null int64
date                 49105 non-null int64
time                 49105 non-null int64
status_of_system     49105 non-null object
latitude             49105 non-null object
longitude            49105 non-null object
max_sustained_wind   49105 non-null int64
central_pressure     18436 non-null float64
dtypes: float64(1), int64(5), object(5)
memory usage: 4.1+ MB
```

In [26]:
```
df.describe()
```

Out[26]:

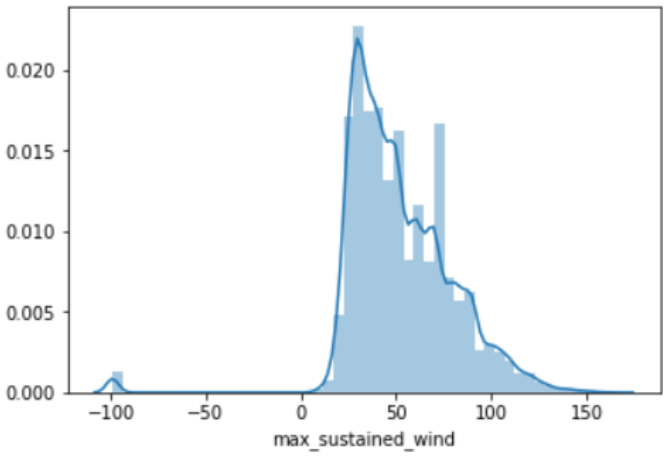
	year	cyclone_of_the_year	date	time	max_sustained_wind	central_pressure
count	49105.000000	49105.000000	4.910500e+04	49105.000000	49105.000000	18436.000000
mean	1949.711944	7.439487	1.949802e+07	910.125975	52.005091	992.244250
std	44.618521	5.226704	4.461850e+05	671.043363	27.681902	19.113748
min	1851.000000	1.000000	1.851062e+07	0.000000	-99.000000	882.000000
25%	1911.000000	3.000000	1.911110e+07	600.000000	35.000000	984.000000
50%	1956.000000	6.000000	1.956093e+07	1200.000000	45.000000	999.000000
75%	1989.000000	10.000000	1.989081e+07	1800.000000	70.000000	1006.000000
max	2015.000000	31.000000	2.015111e+07	2330.000000	165.000000	1024.000000

In [27]:
```
sns.countplot(x="status_of_system", data=df)
plt.show()
```



In [28]:
```
sns.distplot(df["max_sustained_wind"])
```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x2051leab128>



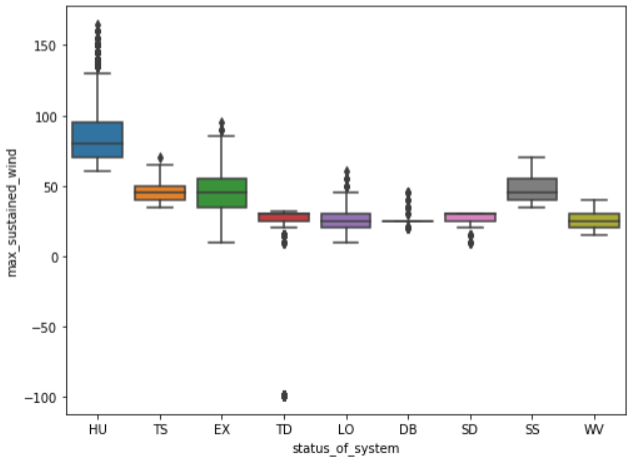
b)
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
df=pd.read_csv("Atlantic.csv")
df['Power']=df['Power'].fillna(df['central_pressure'].mean())
sns.boxplot(x='status_of_system', y='max_sustained_wind', data=df)
```

```
plt.show()

sns.scatterplot(x='central_pressure', y='max_sustained_wind', data=df)
plt.show()
```

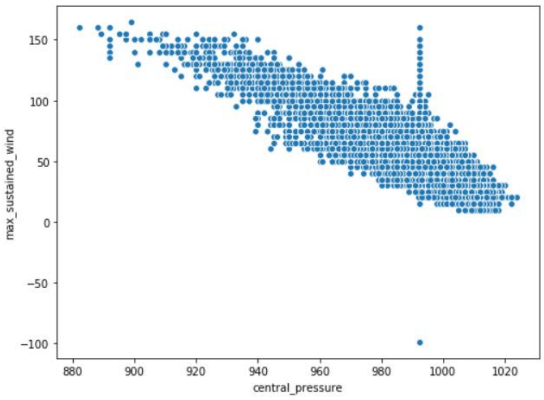
```
In [36]: plt.figure(figsize=(8, 6))
         df['central_pressure']=df['central_pressure'].fillna(df['central_pressure'].mean())
         sns.boxplot(x='status_of_system', y='max_sustained_wind', data=df)
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x20511f948d0>
```




```
In [38]: plt.figure(figsize=(8, 6))
         sns.scatterplot(x='central_pressure', y='max_sustained_wind', data=df)
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x20511dce518>
```



c)

The count of missing values in field central_pressure, so I fill missing values with the mean of central_pressure. This is the most common method. The disadvantage of using mean is that the mean is greatly affected by outliers in our data.

2.

a)
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import csv
csv_row_data = []
for page in range(1,6):
    res = requests.get("https://top500.org/list/2019/06/?page={}".format(page))
    soup = BeautifulSoup(res.content,'lxml')
    all_row_data=soup.select("table tr")
    for idx,row in enumerate(all_row_data):
        if idx == 0:
            continue
        all_tds=row.select("td")
        rank = all_tds[0].text
        print(rank)
        system=all_tds[1].text
        cores=all_tds[3].text.replace(",","")
        rmax=all_tds[4].text.replace(",","")
        rpeak=all_tds[5].text.replace(",","")
        power=all_tds[6].text.replace(",","")
        csv_row_data.append([rank,system,cores,rmax,rpeak,power])
with open("top500.csv","w",encoding="utf-8", newline='') as f:
    f_csv = csv.writer(f)
    f_csv.writerow(["Rank", "System", "Cores", "RMax", "RPeak", "Power"])
    f_csv.writerows(csv_row_data)
```

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import csv
csv_row_data = []
for page in range(1,6):
    res = requests.get("https://top500.org/list/2019/06/?page={}".format(page))
    soup = BeautifulSoup(res.content,'lxml')
    all_row_data=soup.select("table tr")
    for idx,row in enumerate(all_row_data):
        if idx == 0:
            continue
        all_tds=row.select("td")
        rank = all_tds[0].text
        print(rank)
        system=all_tds[1].text
        cores=all_tds[3].text.replace(",","")
        rmax=all_tds[4].text.replace(",","")
        rpeak=all_tds[5].text.replace(",","")
        power=all_tds[6].text.replace(",","")
        csv_row_data.append([rank,system,cores,rmax,rpeak,power])

with open("top500.csv","w",encoding="utf-8", newline='') as f:
    f_csv = csv.writer(f)
    f_csv.writerow(["Rank", "System", "Cores", "RMax", "RPeak", "Power"])
    f_csv.writerows(csv_row_data)
```

b）

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
top500_df=pd.read_csv("top500.csv")
top500_df=top500_df[["Cores","RMax","RPeak","Power"]]
print("summary statistics of Cores, RMax, RPeak, and Power:")
print(top500_df.describe())
top500_df['Power']=top500_df['Power'].fillna(top500_df['Power'].mean())
for col in ["Cores","RMax","RPeak","Power"]:
    sns.distplot(top500_df[col],label=col)
    plt.show()
```

In [44]:
```
top500_df=pd.read_csv("top500.csv")
```

In [45]:
```
top500_df.head()
```

Out[45]:

	Rank	System	Cores	RMax	RPeak	Power
0	1	DOE/SC/Oak Ridge National LaboratoryUnited States	2414592	148600.0	200794.9	10096.0
1	2	DOE/NNSA/LLNLUnited States	1572480	94640.0	125712.0	7438.0
2	3	National Supercomputing Center in WuxiChina	10649600	93014.6	125435.9	15371.0
3	4	National Super Computer Center in GuangzhouChina	4981760	61444.5	100678.7	18482.0
4	5	Texas Advanced Computing Center/Univ. of Texas...	448448	23516.4	38745.9	NaN

In [47]:
```
top500_df.info()
```

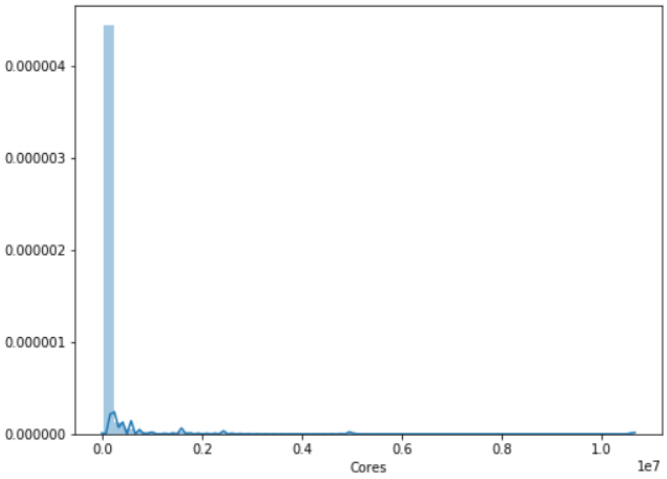
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
Rank       500 non-null int64
System     500 non-null object
Cores      500 non-null int64
RMax       500 non-null float64
RPeak      500 non-null float64
Power      209 non-null float64
dtypes: float64(3), int64(2), object(1)
memory usage: 23.5+ KB
```

In [49]:
```
top500_df=top500_df[["Cores","RMax","RPeak","Power"]]
print("summary statistics of Cores, RMax, RPeak, and Power:")
print(top500_df.describe())
```

```
summary statistics of Cores, RMax, RPeak, and Power:
                Cores           RMax           RPeak          Power
count  5.000000e+02     500.000000     500.000000     209.000000
mean   1.182127e+05    3119.151200    4927.748800    1756.617225
std    5.472871e+05    9556.759821   13282.606456    2584.792937
min    1.259200e+04    1021.000000    1164.700000      81.000000
25%    3.600000e+04    1179.900000    2119.700000     544.000000
50%    5.760000e+04    1646.050000    2404.800000     917.000000
75%    7.570000e+04    1986.650000    3779.200000    1620.000000
max    1.064960e+07  148600.000000  200794.900000   18482.000000
```

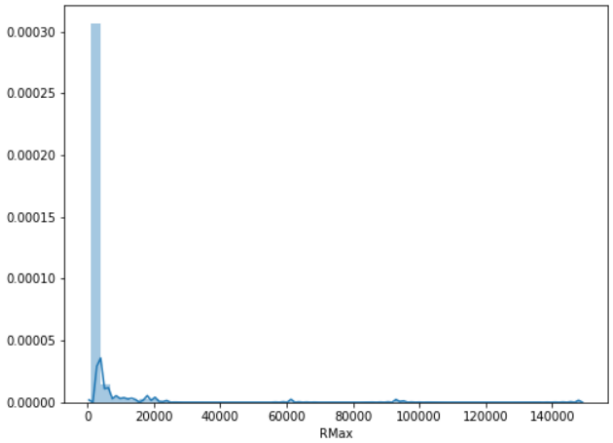
In [52]:
```
plt.figure(figsize=(8, 6))
sns.distplot(top500_df["Cores"], label=col)
```

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x20513ded7f0>



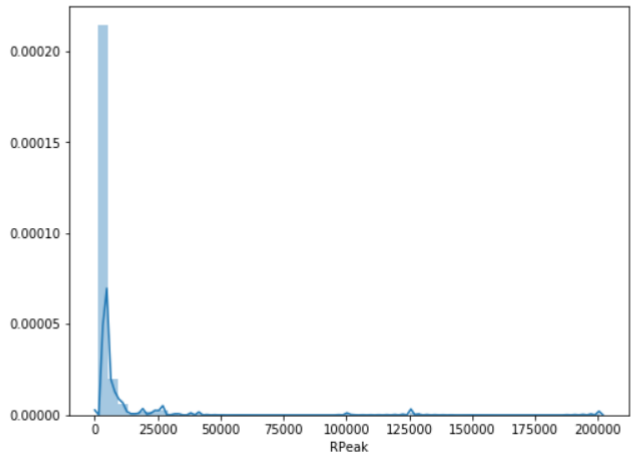
In [53]:
```
plt.figure(figsize=(8, 6))
sns.distplot(top500_df["RMax"],label=col)
```

Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x20513ea70b8>



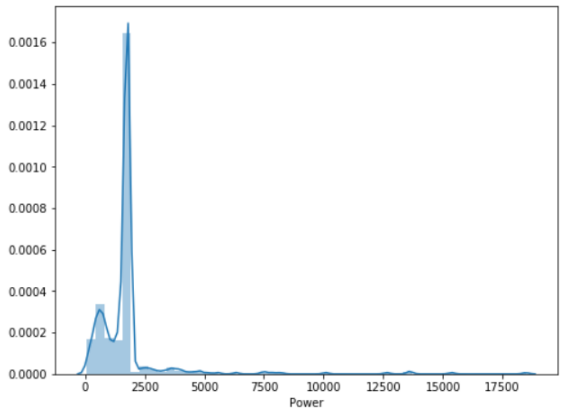
In [54]:
```
plt.figure(figsize=(8, 6))
sns.distplot(top500_df["RPeak"],label=col)
```

Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x20513f7f160>



In [56]:
```
plt.figure(figsize=(8, 6))
top500_df['Power']=top500_df['Power'].fillna(top500_df['Power'].mean())
sns.distplot(top500_df["Power"],label=col)
```

Out[56]: <matplotlib.axes._subplots.AxesSubplot at 0x20511dce2b0>

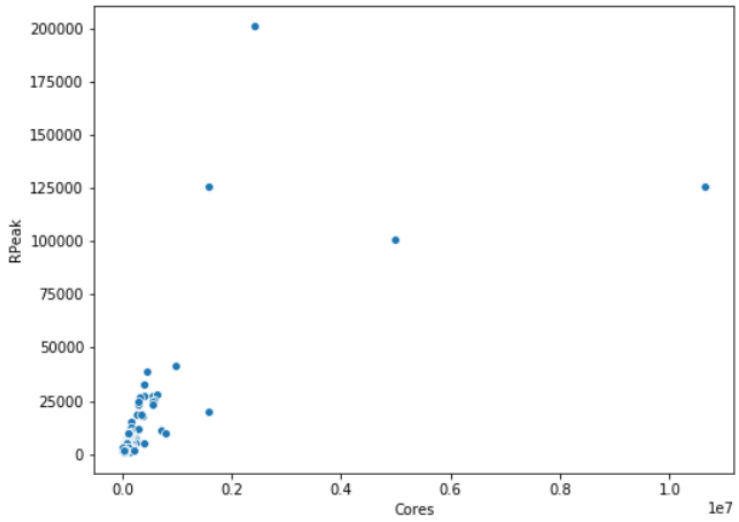


c)
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
top500_df=pd.read_csv("top500.csv")
sns.scatterplot(x='Cores',y='RPeak',data=top500_df)
plt.show()

sns.scatterplot(x='Cores',y='Power',data=top500_df)
plt.show()
```

In [58]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Cores',y='RPeak',data=top500_df)
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x20513b1fac8>



In [59]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Cores',y='Power',data=top500_df)
```

Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x2051428d198>