

In [33]:

```

from __future__ import print_function, division

import thinkdsp
import thinkplot
import thinkstats2

import numpy as np
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets

%matplotlib inline

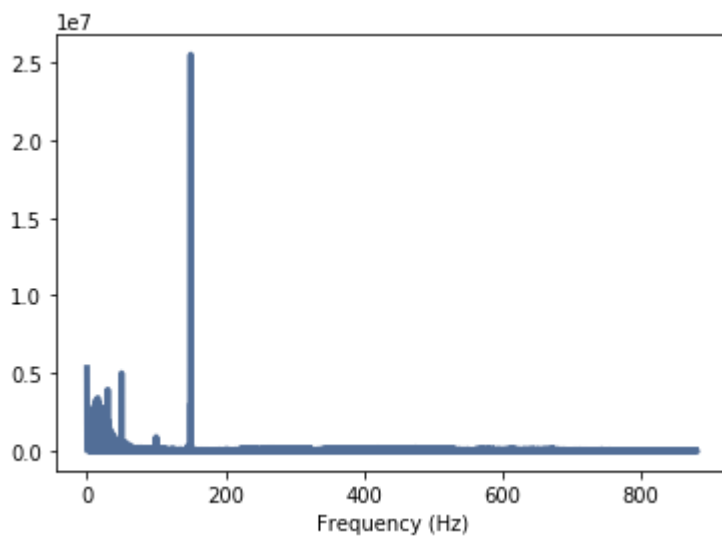
```

In [34]:

```

wave=thinkdsp.read_wave('56611__inchadney__morning-in-the-country.wav')
spectrum=wave.make_spectrum()
spectrum.plot_power(high=880)
thinkplot.config(xlabel='Frequency (Hz)')

```



In [35]:

```

result = spectrum.estimate_slope()#기울기 측정
result.slope#기울기 즉 베타가 절대값 20이상이면 Brownian noise입니다.

```

Out[35]:

-2.953502435683652

In [36]:

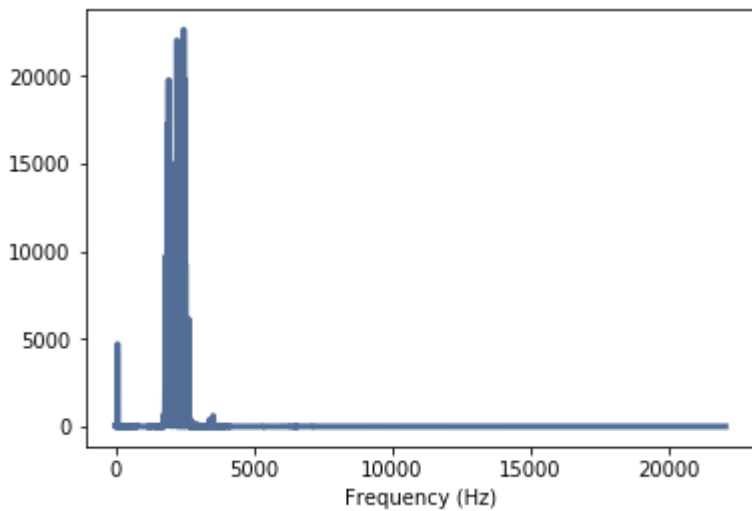
```
segment=wave.segment(start=1,duration=1)
segment.make_audio()
```

Out[36]:

0:00 / 0:01

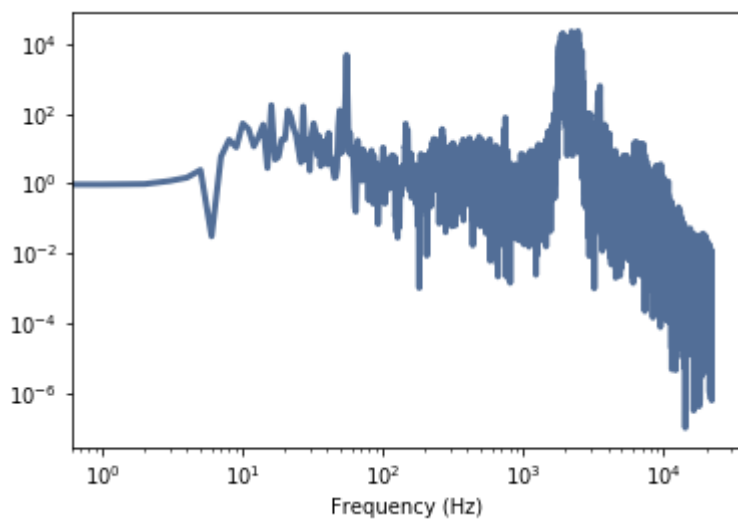
In [37]:

```
spectrum=segment.make_spectrum()
spectrum.plot_power()
thinkplot.config(xlabel='Frequency (Hz)')
```



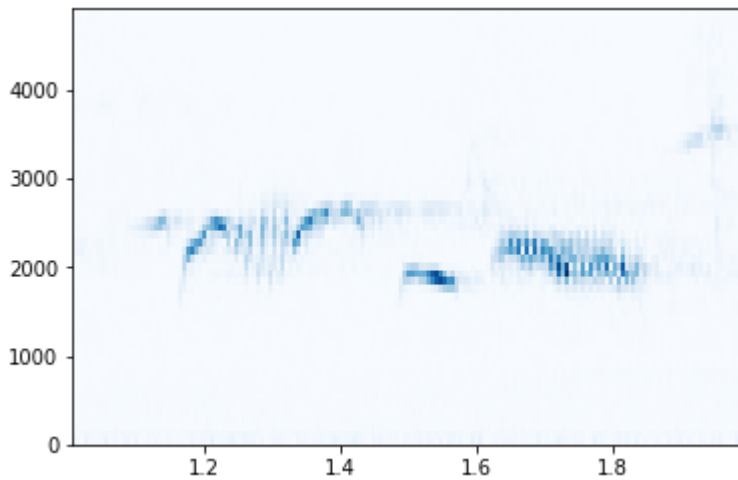
In [38]:

```
spectrum.plot_power()#첫번째 파일을 Log-log scale로 그리기
thinkplot.config(xlabel='Frequency (Hz)',
                  xscale='log',
                  yscale='log')
```



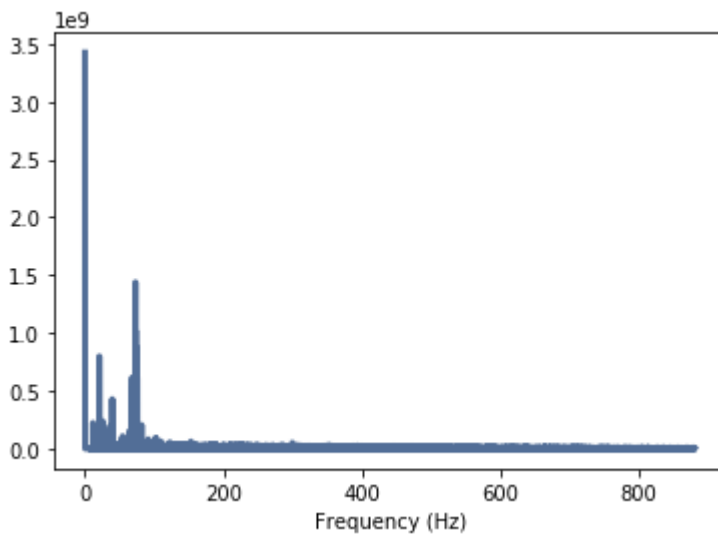
In [39]:

```
segment.make_spectrogram(512).plot(high=5000)
```



In [40]:

```
wave2=thinkdsp.read_wave('13793_soarer__north-sea.wav')
spectrum2=wave2.make_spectrum()
spectrum2.plot_power(high=880)
thinkplot.config(xlabel='Frequency (Hz)')
```



In [41]:

```
result = spectrum2.estimate_slope()#기울기 측정
result.slope#기울기가 절대값 2에 수렴하므로 Brownian noise입니다.
```

Out[41]:

```
-2.0885624235894764
```

In [42]:

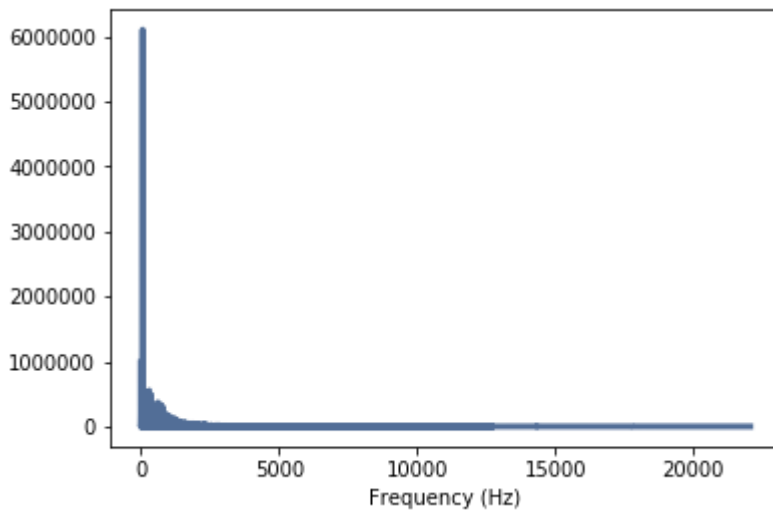
```
segment2=wave2.segment(start=1,duration=5)
segment2.make_audio()
```

Out[42]:

0:00 / 0:05

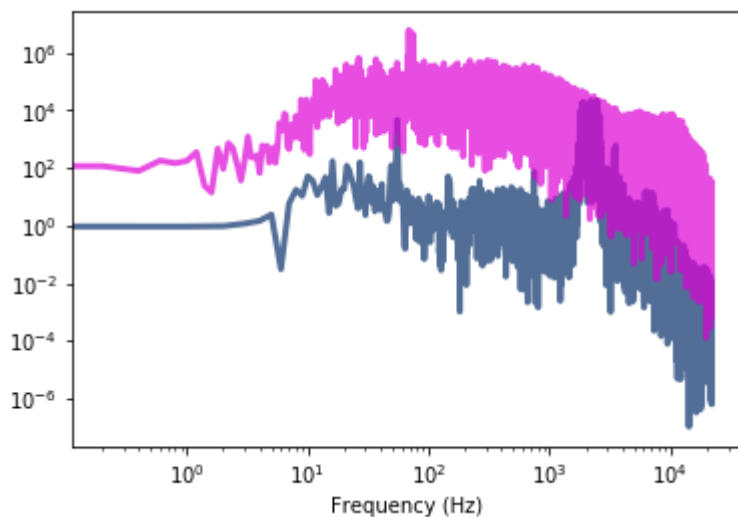
In [43]:

```
spectrum2=segment2.make_spectrum()
spectrum2.plot_power()#두번째 파일의 noise 그리기
thinkplot.config(xlabel='Frequency (Hz)')
```



In [44]:

```
spectrum.plot_power()
spectrum2.plot_power(color='#DF01D7')
thinkplot.config(xlabel='Frequency (Hz)',
                  xscale='log',
                  yscale='log')
```



In []: