In [30]:

```
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')

%matplotlib inline
```

In [31]:

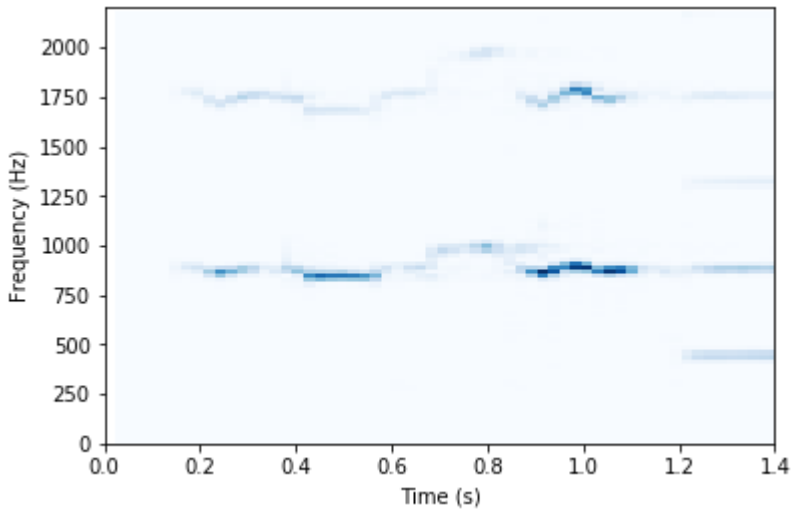
```
wave=thinkdsp.read_wave('92002__jcveliz__violin-origional.wav')
wave.normalize()
wave.make_audio()
```

Out[31]:

0:00 / 0:05

In [44]:

```
wave.make_spectrogram(2048).plot(high=2200)
thinkplot.config(xlabel='Time (s)',
                 ylabel='Frequency (Hz)',
                 xlim=[0, 1.4],
                 ylim=[0, 2200])
```



In [32]:

```
from autocorr import autocorr
```

In [33]:

```
def estimate_fundamental(segment, low=70, high=150):
    lags, corrs = autocorr(segment)
    lag = np.array(corrs[low:high]).argmax() + low #argmax()함수는 제일 큰 값의 인덱스를 return
    period = lag / segment.framerate
    frequency = 1 / period
    return frequency
```

In [34]:

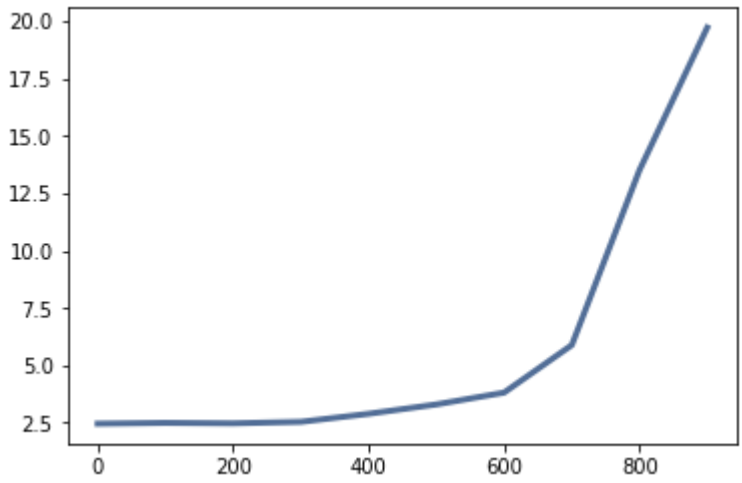
```
duration = 0.01
segment = wave.segment(start=0.2, duration=duration)
freq = estimate_fundamental(segment)
freq
```

Out[34]:

432.35294117647055

In [35]:

```
duration = 0.01
segment = wave.segment(start=0.2, duration=duration)
spectrum=segment.make_spectrum()
spectrum.plot(high=1000)
```



In [36]:

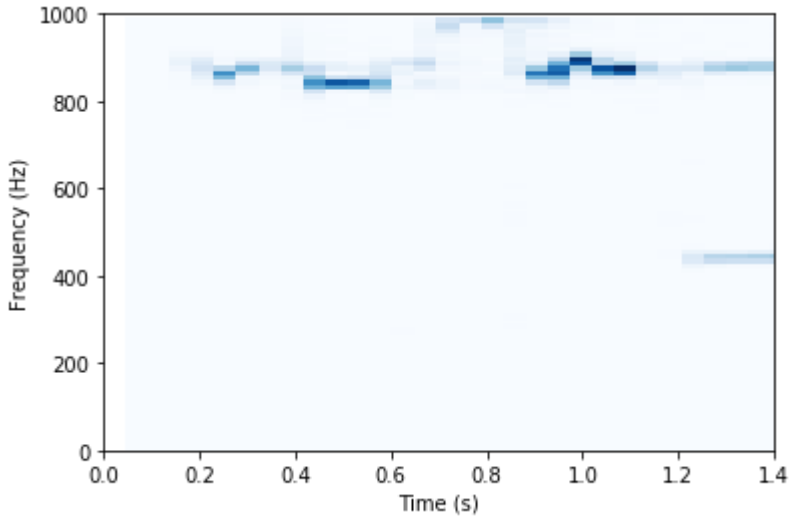
```
step = 0.05
starts = np.arange(0.0, 1.4, step)

ts = []
freqs = []

for start in starts:
    ts.append(start + step/2)
    segment = wave.segment(start=start, duration=duration)
    freq = estimate_fundamental(segment)
    freqs.append(freq)
```

In [45]:

```
wave.make_spectrogram(4096).plot(high=1000)
thinkplot.config(xlabel='Time (s)',
                 ylabel='Frequency (Hz)',
                 xlim=[0, 1.4],
                 ylim=[0, 1000])
```



In [ ]:

```
```