HW05 B0829024

Exercise 02

In [2]:

```
# Get thinkdsp.py

import os

if not os.path.exists('thinkdsp.py'):
    !wget https://github.com/AllenDowney/ThinkDSP/raw/master/code/thinkdsp.py
```

```
--2022-04-13 11:48:16--  https://github.com/AllenDowney/ThinkDSP/raw/master/code/thinkdsp
.py
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/code/thinkdsp.py
[following]
--2022-04-13 11:48:16--  https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/co
de/thinkdsp.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.1
99.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443.
.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 48687 (48K) [text/plain]
Saving to: 'thinkdsp.py'

thinkdsp.py         100%[===================>]  47.55K  --.-KB/s    in 0.01s

2022-04-13 11:48:16 (4.67 MB/s) - 'thinkdsp.py' saved [48687/48687]
```

In [3]:

```
import numpy as np
import matplotlib.pyplot as plt

from thinkdsp import decorate
```

In [4]:

```
def serial_corr(wave, lag=1):
    n = len(wave)
    y1 = wave.ys[lag:]
    y2 = wave.ys[:n-lag]
    corr = np.corrcoef(y1, y2, ddof=0)[0, 1]
    return corr
```

In [5]:

```
def autocorr(wave):
    lags = range(len(wave.ys)//2)
    corrs = [serial_corr(wave, lag) for lag in lags]
    return lags, corrs
```

In [6]:

```
def estimate_fundamental(wave,frag_len):
  wave_len = len(wave.ys)//wave.framerate
  freqs = []
  for i in range(int(wave_len//frag_len)):
    sub_wave = wave.segment(start=i*frag_len, duration=frag_len)
```

```
    lags,corrs = autocorr(sub_wave)
    # plt.plot(lags, corrs)
    # decorate(xlabel='Lag (index)', ylabel='Correlation')
    low, high = 50, 150
    lag = np.array(corrs[low:high]).argmax() + low
    # lag = np.array(corrs).argmax()
    period = lag / sub_wave.framerate
    frequency = 1 / period
    freqs.append(frequency)
  return freqs
```

In [7]:

```
if not os.path.exists('28042__bcjordan__voicedownbew.wav'):
    !wget https://github.com/AllenDowney/ThinkDSP/raw/master/code/28042__bcjordan__voicedownbew.wav
```

```
--2022-04-13 11:48:17--  https://github.com/AllenDowney/ThinkDSP/raw/master/code/28042__bcjordan__voicedownbew.wav
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/code/28042__bcjordan__voicedownbew.wav [following]
--2022-04-13 11:48:17--  https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/code/28042__bcjordan__voicedownbew.wav
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 125996 (123K) [audio/wav]
Saving to: '28042__bcjordan__voicedownbew.wav'

28042__bcjordan__vo 100%[===================>] 123.04K  --.-KB/s    in 0.02s

2022-04-13 11:48:17 (5.53 MB/s) - '28042__bcjordan__voicedownbew.wav' saved [125996/125996]
```

In [8]:

```
from thinkdsp import read_wave

wave = read_wave('28042__bcjordan__voicedownbew.wav')
wave.normalize()
wave.make_audio()
```

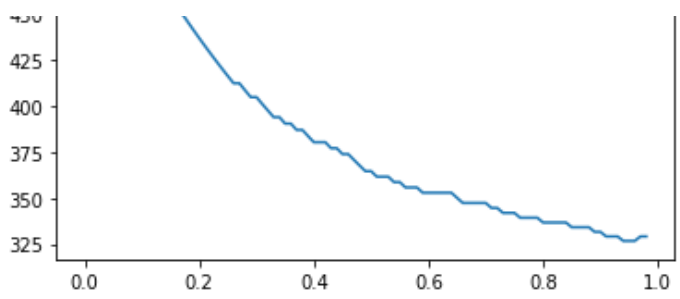
Out[8]:

Your browser does not support the audio element.

In [9]:

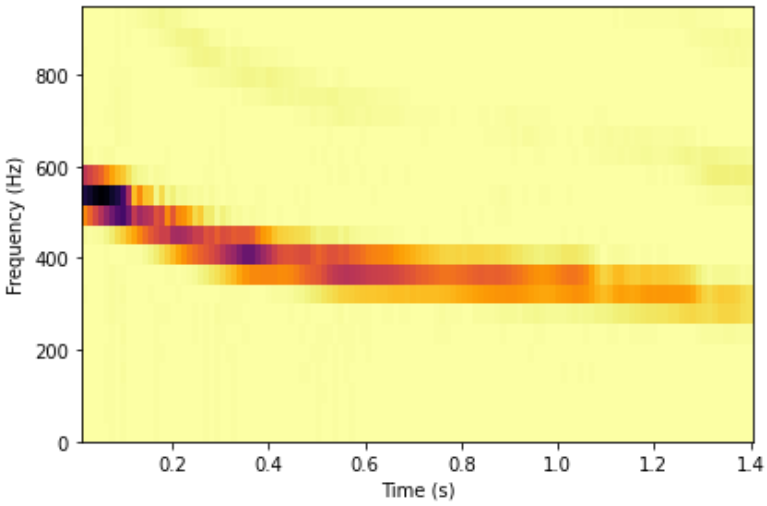
```
duration = 0.01
# segment = wave.segment(start=0.2, duration=duration)
freqs = estimate_fundamental(wave,duration)
t = [i*duration for i in range(len(freqs))]
# print(freqs)
plt.plot(t,freqs)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: bias and ddof have no effect and are deprecated
  """
```



In [10]:

```
spectro = wave.make_spectrogram(seg_length=1024)
spectro.plot(high=1000)
decorate(xlabel='Time (s)',
                 ylabel='Frequency (Hz)')
```



上述音樂為freq從500HZ到300HZ的嗚啾聲(與課本5.4相同)，可以看到estimate_fundamental function可以正確地繪製出頻率圖。

Exercise 03

In [11]:

```
if not os.path.exists('BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv'):
    !wget https://github.com/AllenDowney/ThinkDSP/raw/master/code/BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv
```

```
--2022-04-13 12:03:41--  https://github.com/AllenDowney/ThinkDSP/raw/master/code/BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/code/BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv [following]
--2022-04-13 12:03:42--  https://raw.githubusercontent.com/AllenDowney/ThinkDSP/master/code/BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 143622 (140K) [text/plain]
Saving to: 'BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv'

BTC_USD_2013-10-01_ 100%[===================>] 140.26K  --.-KB/s    in 0.02s

2022-04-13 12:03:42 (6.24 MB/s) - 'BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv' saved [143622/143622]
```

In [12]:

```
import pandas as pd

df = pd.read_csv('BTC_USD_2013-10-01_2020-03-26-CoinDesk.csv',
                 parse_dates=[0])
df
```

Out[12]:

	Currency	Date	Closing Price (USD)	24h Open (USD)	24h High (USD)	24h Low (USD)
0	BTC	2013-10-01	123.654990	124.304660	124.751660	122.563490
1	BTC	2013-10-02	125.455000	123.654990	125.758500	123.633830
2	BTC	2013-10-03	108.584830	125.455000	125.665660	83.328330
3	BTC	2013-10-04	118.674660	108.584830	118.675000	107.058160
4	BTC	2013-10-05	121.338660	118.674660	121.936330	118.005660
...	...	...	...	...	...	...
2354	BTC	2020-03-22	5884.340133	6187.042146	6431.873162	5802.553402
2355	BTC	2020-03-23	6455.454688	5829.352511	6620.858253	5694.198299
2356	BTC	2020-03-24	6784.318011	6455.450650	6863.602196	6406.037439
2357	BTC	2020-03-25	6706.985089	6784.325204	6981.720386	6488.111885
2358	BTC	2020-03-26	6721.495392	6697.948320	6796.053701	6537.856462

2359 rows × 6 columns

In [17]:

```
from thinkdsp import Wave
ys = df['Closing Price (USD)']
ts = df.index
wave = Wave(ys, ts, framerate=1)
```

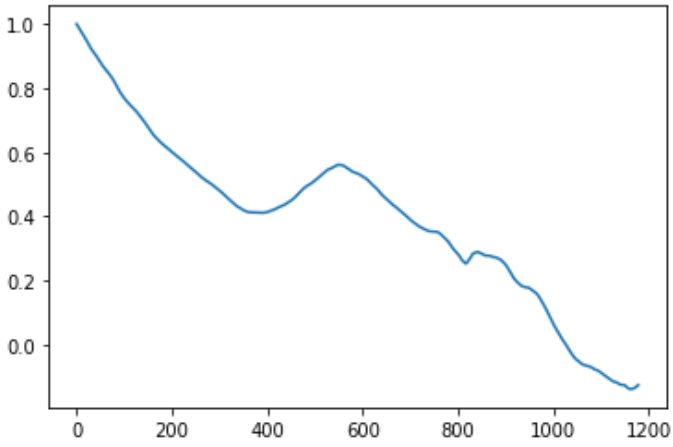
In [14]:

```
lags, corrs = autocorr(wave)
plt.plot(lags, corrs)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: bias
and ddof have no effect and are deprecated
  """
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x7f47a9d1cf50>]
```



In [18]:

```
low, high = 400,600
```

```
lag = np.array(corrs[low:high]).argmax() + low
print('lag=',lag)
period = lag / wave.framerate
frequency = 1 / period
print(frequency)
```

```
lag= 551
0.0018148820326678765
```

上圖可以看到比特幣的價格在自相關性上，隨著時間的增加勻速下降。也就是距離天數越遠價格並不持續相關。如果在中遠時間想要找到最相關的距離，則551天的價格走勢與第0天最為相似，可以達到將近0.6的正相關。

hw04作業討論過，貨幣/股票的價格走勢類似於布朗噪聲，因此相關性在時間拉長的狀態，仍然有一定的相似性。