# An Exploration of Synth Saxophones Using the Frequency Domain

Riley Elliott

2024-05-02

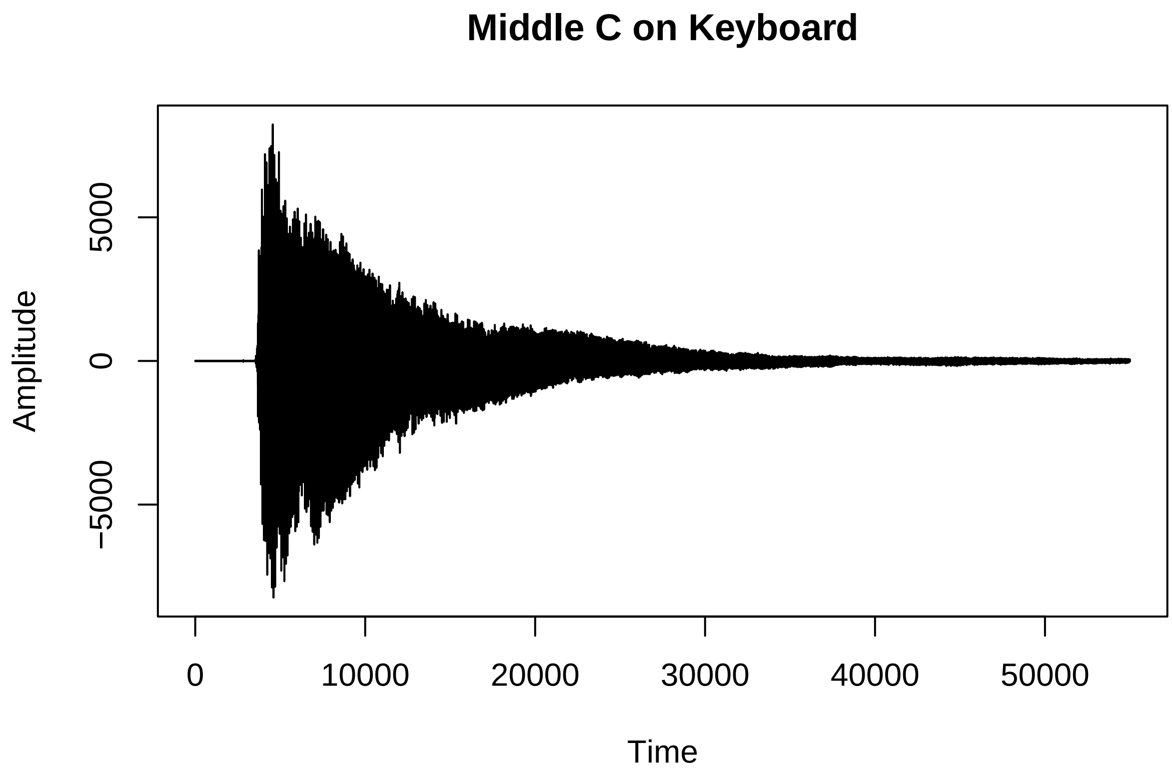
## Introduction

In many ways, music is a very mathematical art. While a lot of things go into making music sound good in the subjective sense, we know that the thing that makes notes sound consonant or dissonant together is the relationship between the frequencies of the sound waves they create. More appropriately, the frequencies *represented* in the sound waves they create. Each note vibrates at multiple frequencies, which are referred to as the harmonic series. The "fundamental frequency" makes the strongest contribution to the sound wave, and a particular set of octaves, fifths, and thirds above it make progressively weaker contributions.

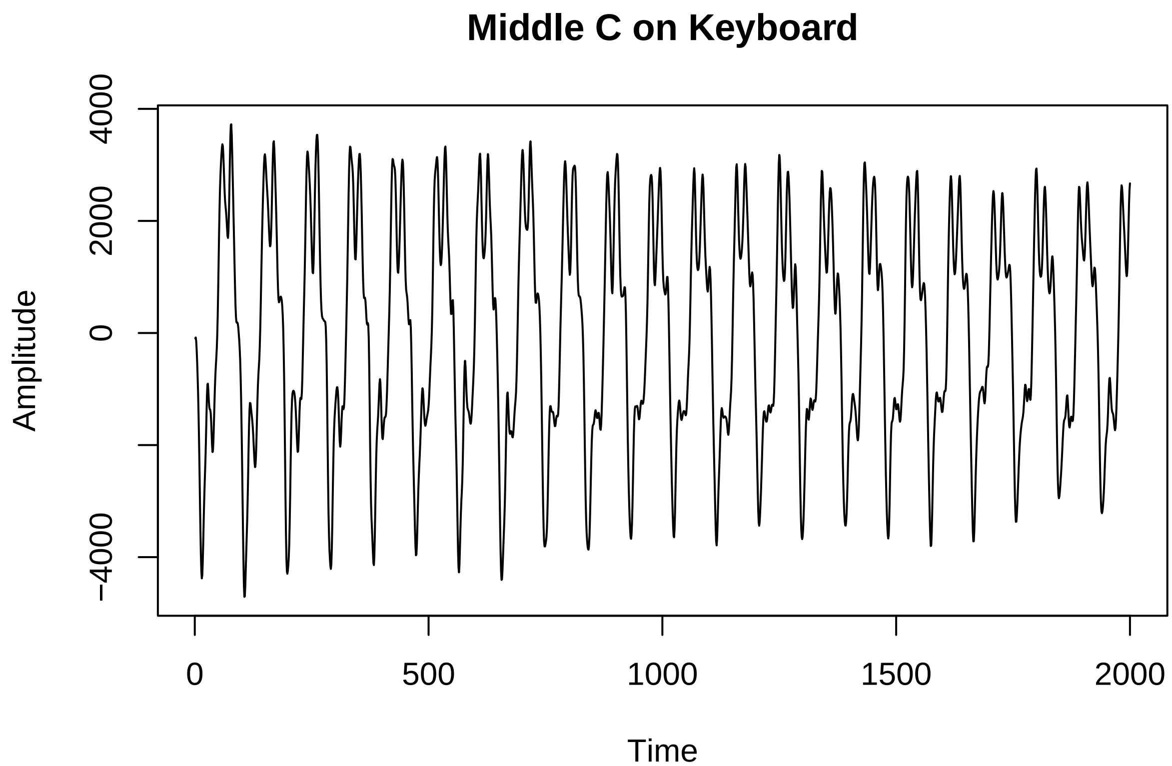
Each instrument is tuned differently–most notably, pianos and keyboards, which are tuned using the equal temperament system, which ensures that one can play in every key but only approximately in tune. Each instrument also has a slightly different harmonic series in that different overtone frequencies are represented at different strengths from instrument to instrument. With that in mind, my goal in this project is to create something of a synth saxophone by transforming the frequency structure of a time series produced by a keyboard, into something more akin to one produced by an alto saxophone. I'll just be doing for one note, to keep things simple.

Here is the time series produced by a middle C on my keyboard.

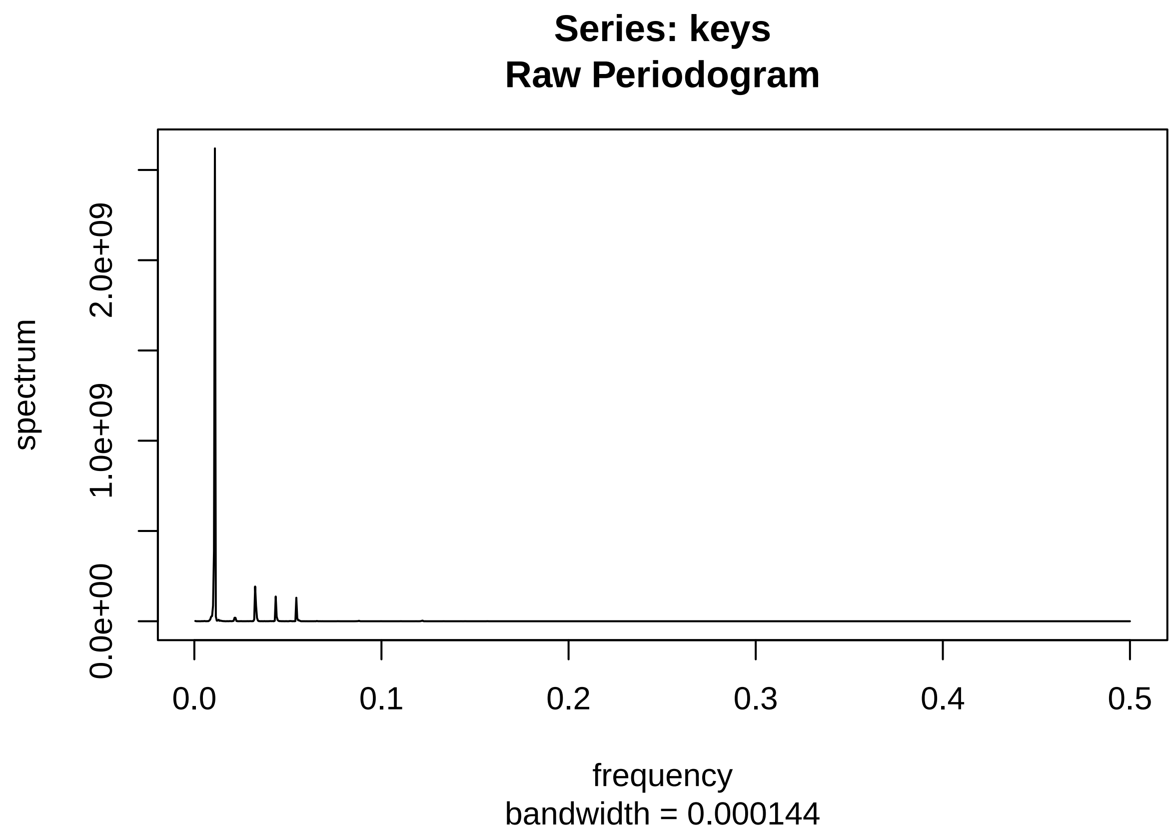
```
##
## Wave Object
##  Number of Samples:      287680
##  Duration (seconds):     11.99
##  Samplingrate (Hertz):   24000
##  Channels (Mono/Stereo): Mono
##  PCM (integer format):   TRUE
##  Bit (8/16/24/32/64):    16
```




I don't have a sustain pedal, so the volume noticeably fades throughout. This makes the time series non-stationary overall. We just need to look at a small enough time window in which the variance doesn't change too much to recover stationarity.



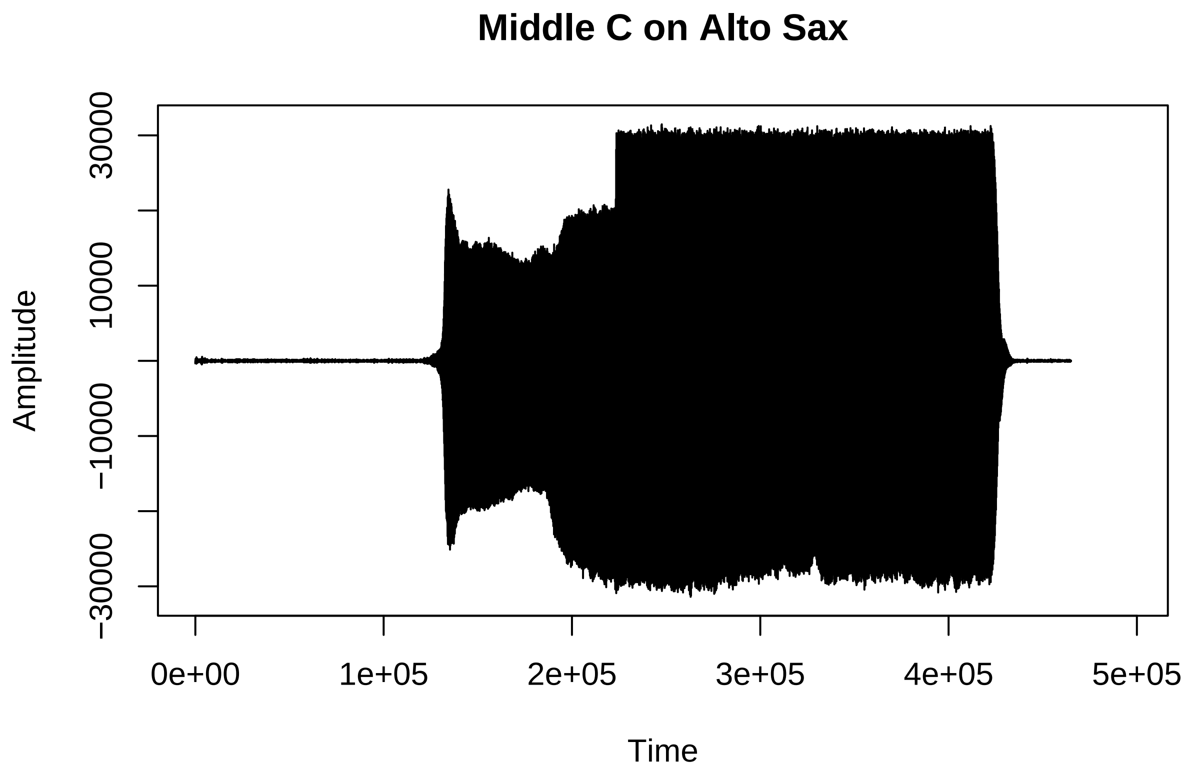

We can see some distinct low frequency structure, and weaker high frequency structure. The time series seems to be very consistent, so there likely aren't many frequencies represented. This makes sense for a musical note–the more frequencies involved, the more muddy and dissonant it will sound. Now, let's plot the periodogram.

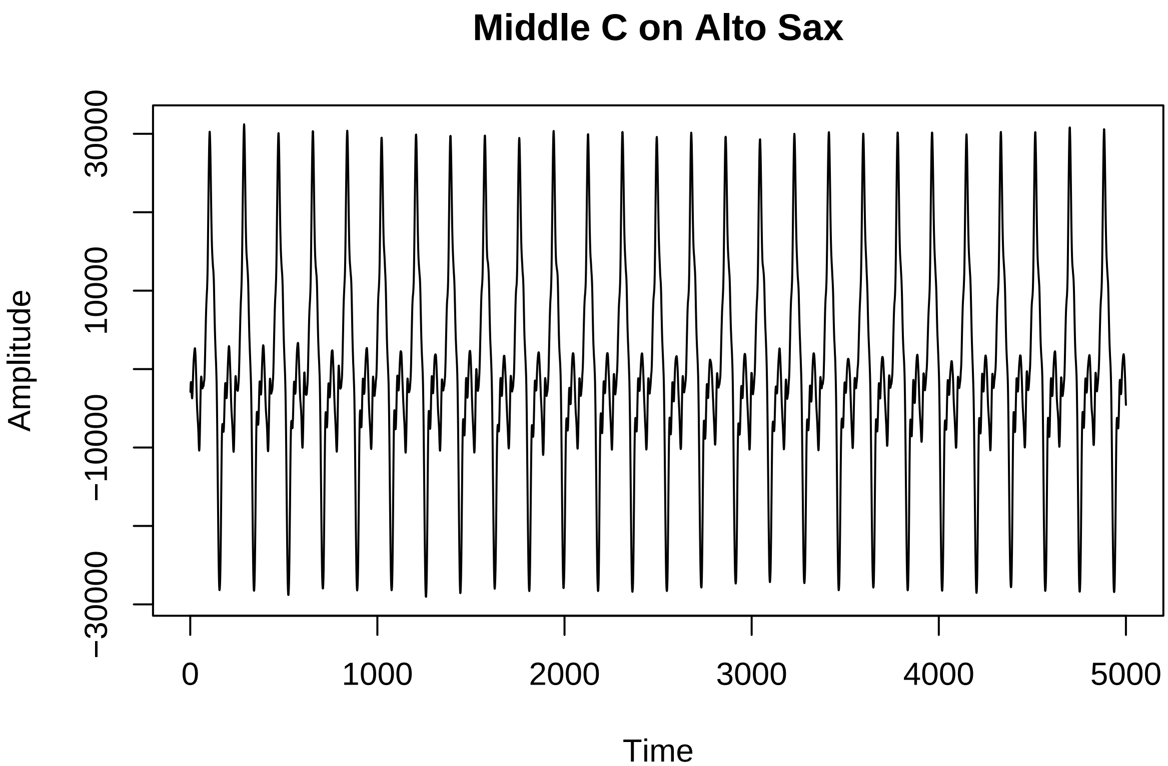


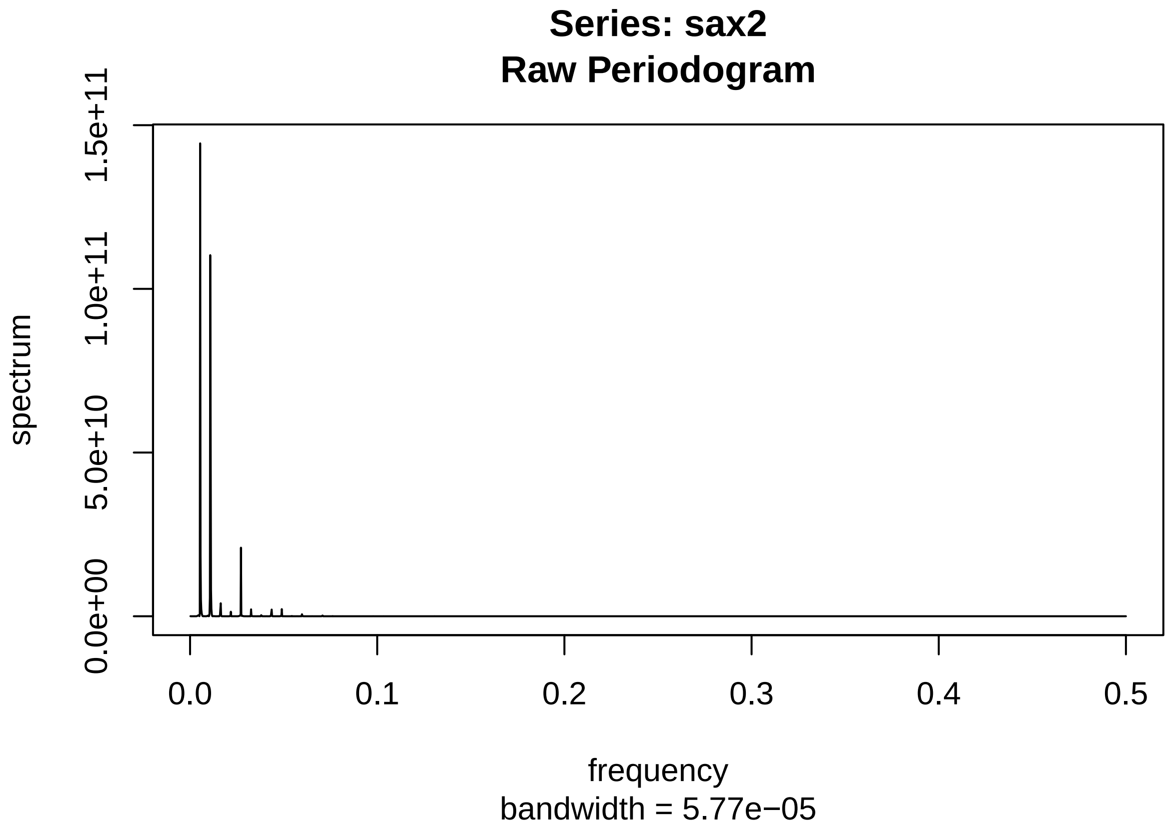

Our suspicions are confirmed. We won't smooth this periodogram for the reasons mentioned above–there is no need to take a local average, because middle c is produced by a very particular combination of a few specific frequencies.

Now let's replicate these plots for the same note played on alto sax.

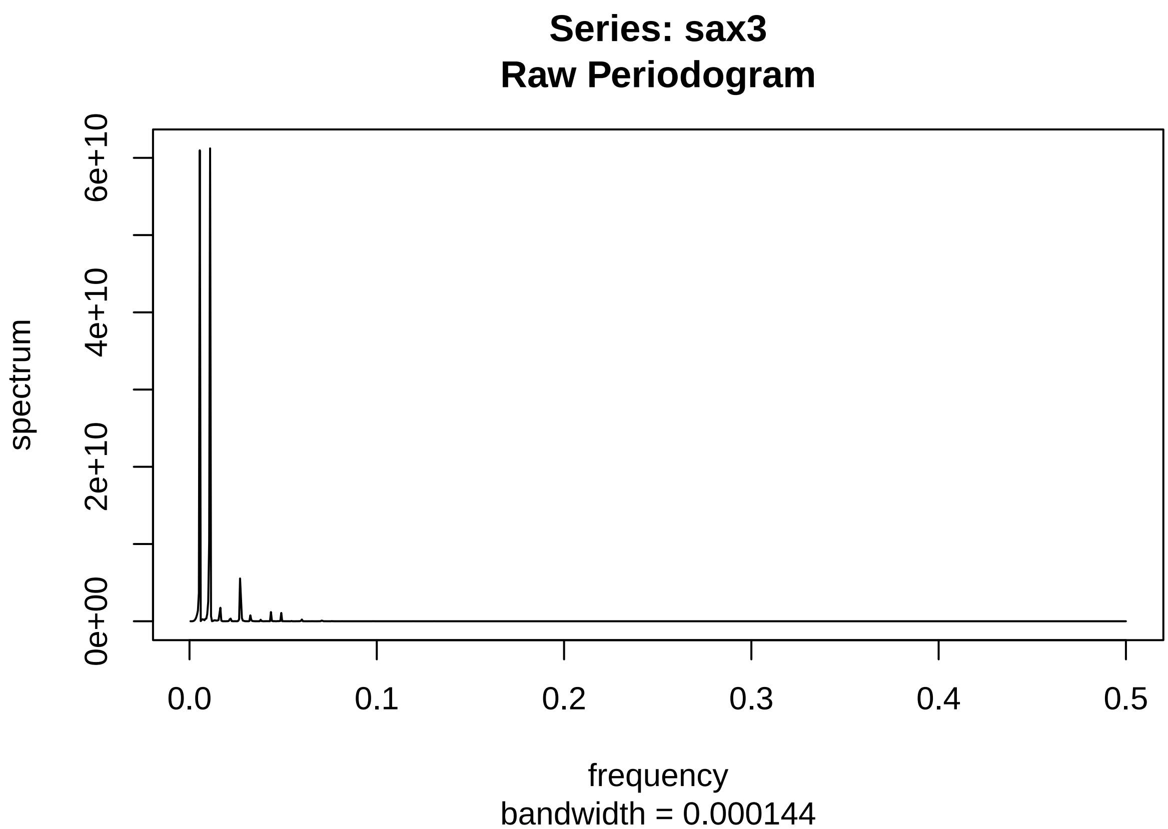
```
##
## Wave Object
##  Number of Samples:      496576
##  Duration (seconds):     10.35
##  Samplingrate (Hertz):   48000
##  Channels (Mono/Stereo): Mono
##  PCM (integer format):   TRUE
##  Bit (8/16/24/32/64):    16
```


Middle C on Alto Sax
Amplitude
30000
10000
−10000
−30000
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Time


Middle C on Alto Sax
Amplitude
30000
10000
−10000
−30000
0
1000
2000
3000
4000
5000
Time




The most notable difference is that the second harmonic, one octave up from the fundamental frequency, is very strongly represented. This might make sense in contrast to a keyboard, which is somewhat mechanical in its production of notes, and can more easily dampen the harmonic series than I can. There are more frequencies represented in general, both for this fact and because my pitch definitely wavered a little bit. I might have even snuck some vibrato in. Both would change the fundamental frequency, and therefore the harmonic series as well.




We're going to reproduce the same spectrum, but with T = 2,000, to match that of the keyboard. This is necessary in order to do the following calculations, in which we attempt to turn $x_t$ into $y_t$ using a linear filter.

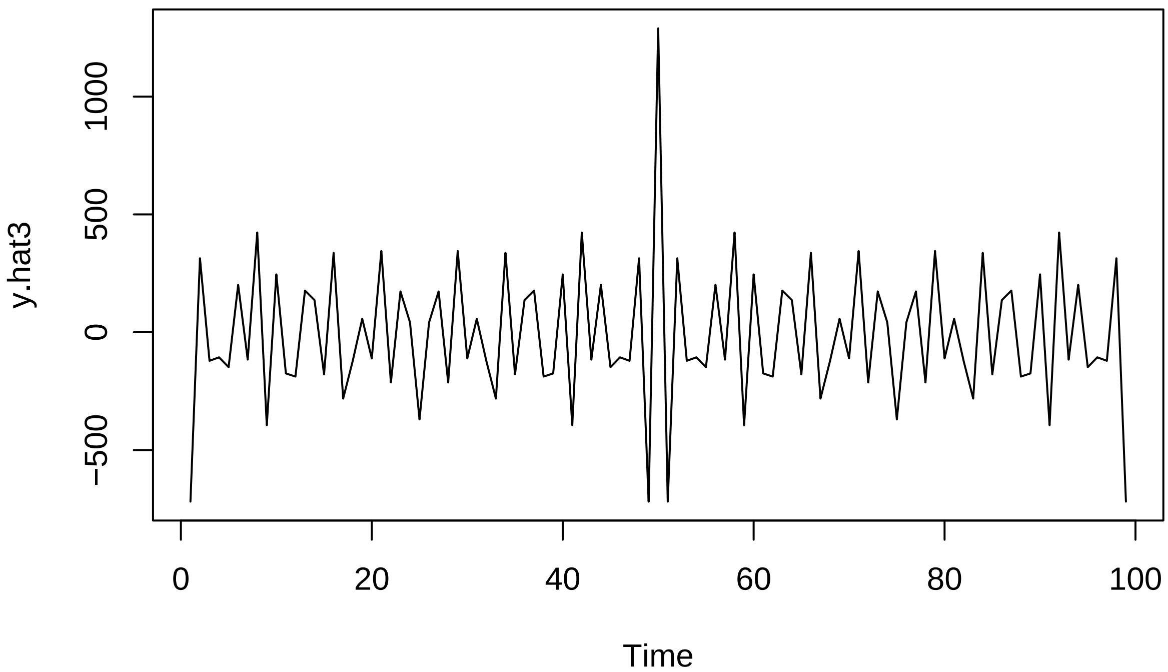
① $d^*(\omega) = \sum_{t=0}^{T} X_t e^{-2\pi i \omega t}$ Discrete fourier transform of $X_t$ (keyboard series) returns $d^*(\omega)$.

② Therefore, $\int_{-1/2}^{1/2} d^*(\omega) e^{2\pi i \omega t} \, d\omega = X_t$. The inverse fourier transform of $d^*(\omega)$ returns $X_t$

③ Let $\hat{Y}_t = \int_{-1/2}^{1/2} d^*(\omega) A(\omega) e^{2\pi i \omega t} \, d\omega$ which must be true for some $A(\omega)$ filter coefficient frequency response series that accentuates some frequencies and dampens others in order to $X_t \to Y_t$ (the sax series).

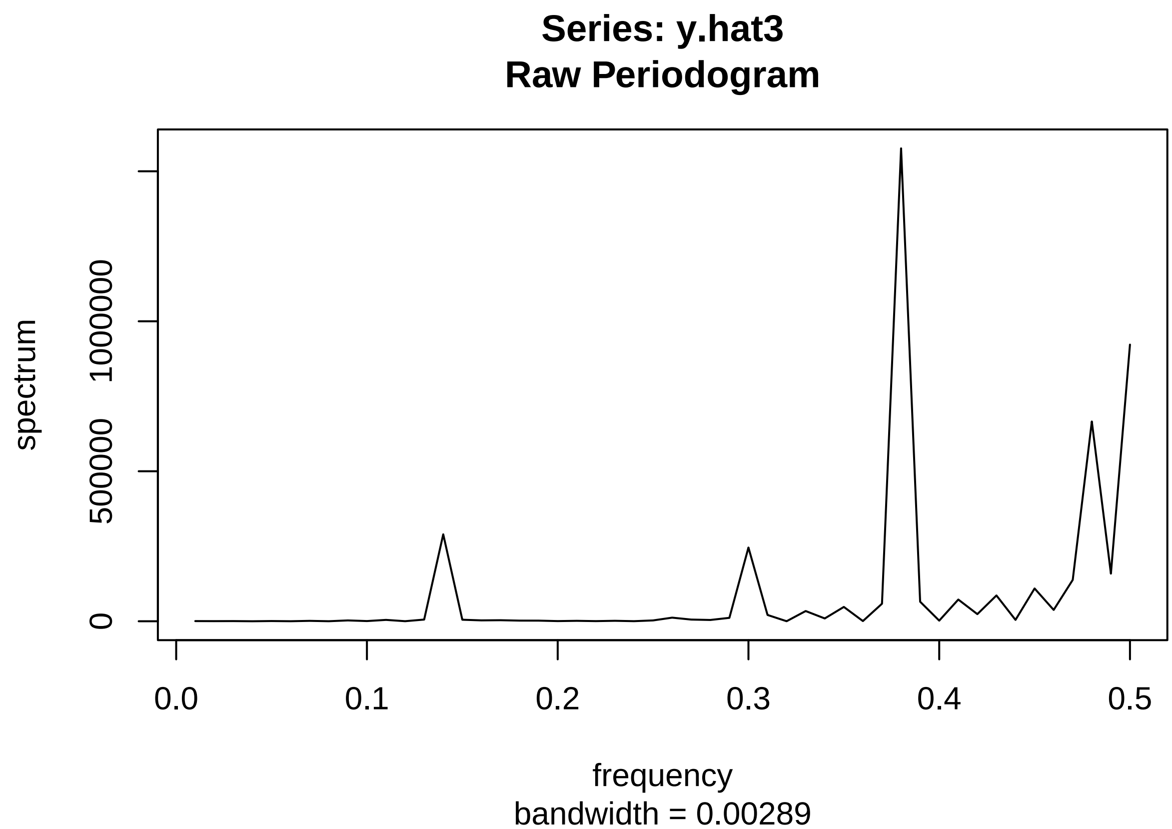
The issue: $|A(\omega)|^2 = \frac{f_Y(\cdot)}{f_X(\cdot)}$, so $A(\omega)$ does not necessarily equal $\frac{f_Y(\omega)}{f_X(\omega)}$. Only if $A(\omega)$ is symmetric, and the complex sin components in Euler's formula cancel, is this true. But we must assume it for our calculations.

Below, $\int_{-1/2}^{1/2} d^*(\omega) \frac{f_Y(\omega)}{f_X(\omega)} e^{2\pi i \omega t} \, d\omega$ is computed in order to return $\hat{y}_t$.




I've forced out the complex components of the results. It's pretty apparent from looking at this time series that it looks nothing like the alto sax time series.

```
spec.pgram(y.hat3, log = "no")
```




Looks like the assumptions underlying our calculations were incorrect. A different method is needed.

I'll attempt to recreate the alto saxophone time series entirely from scratch. In order to do this, I'll need to look through the spectral distribution of the alto to find the most prevalent frequencies and their corresponding strengths. I'll then create a composite waveform by adding cosines of these frequencies with properly weighted amplitudes. First, to find the frequencies.

```
##  [1] 0.0054 0.0056 0.0058 0.0060 0.0106 0.0108 0.0110 0.0112 0.0114 0.0162
## [11] 0.0164 0.0218 0.0272 0.0326 0.0434 0.0436 0.0490
```

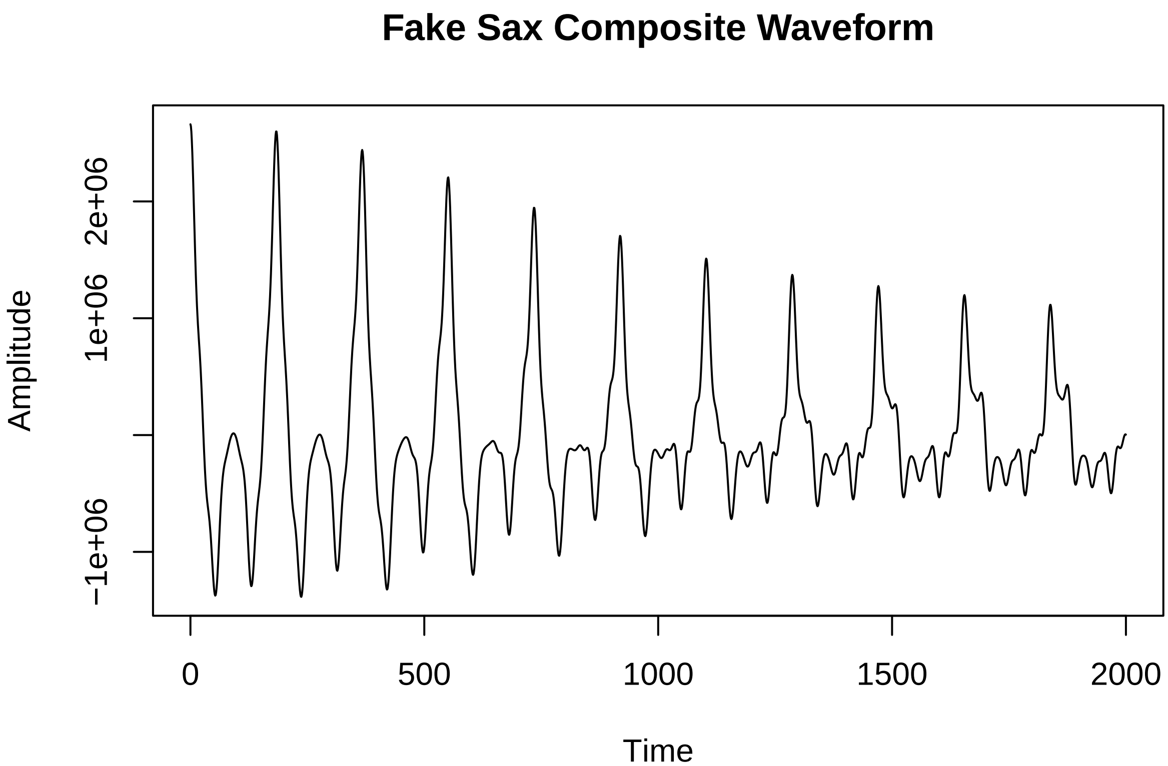
Here, we can see the fundamental frequency is around 0.0054. I kept lowering the minimum required power for a frequency to appear on this list until I could see the third, fifth, and octave were all represented. In most cases, each harmonic is accompanied by a few very similar frequencies, which was not the case with the very precise keyboard. We'll keep those, as they should help provide some sax-like tone. Let's find the corresponding powers of these frequencies.

```
## $`0.0054`
## [1] 144450738897
##
## $`0.0056`
## [1] 18432599286
##
## $`0.0058`
## [1] 5091177345
##
## $`0.006`
```

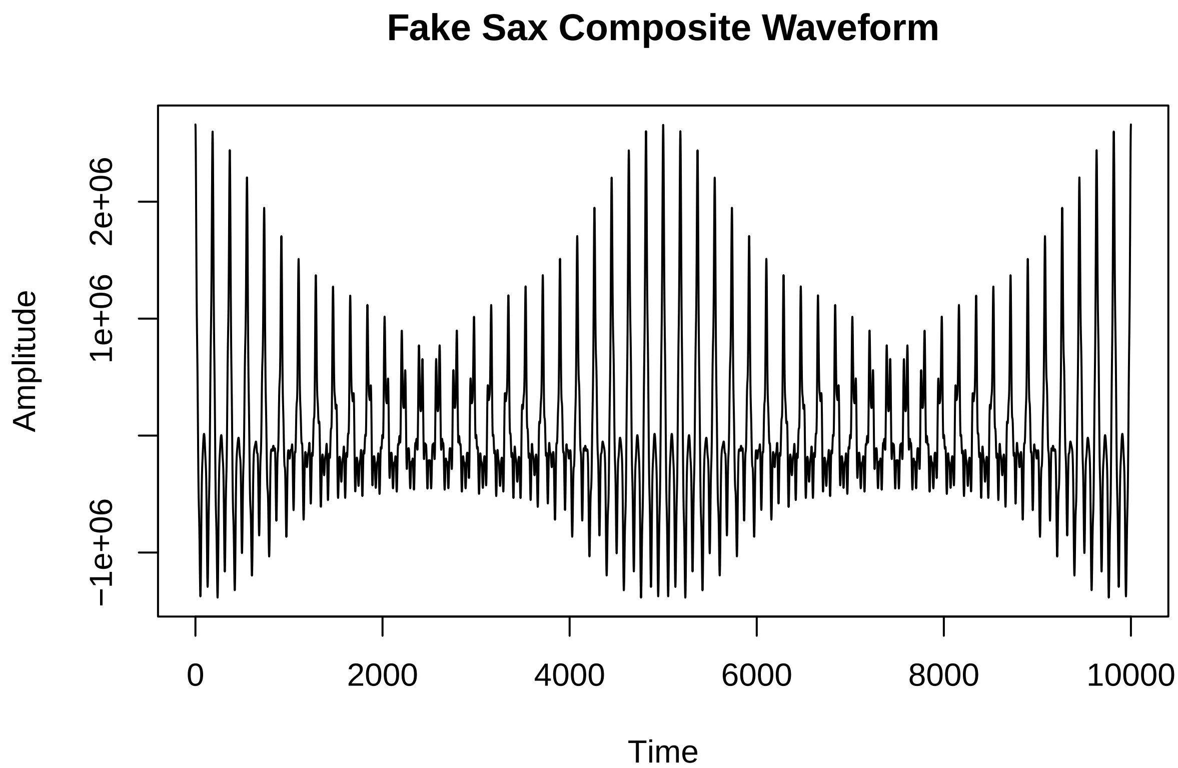
```
## [1] 2162336491
##
## $`0.0106`
## [1] 5681725133
##
## $`0.0108`
## [1] 110295216378
##
## $`0.011`
## [1] 57224615027
##
## $`0.0112`
## [1] 8218549980
##
## $`0.0114`
## [1] 2297621643
##
## $`0.0162`
## [1] 2204821546
##
## $`0.0164`
## [1] 3960610945
##
## $`0.0218`
## [1] 1379491428
##
## $`0.0272`
## [1] 20945750092
##
## $`0.0326`
## [1] 2104587890
##
## $`0.0434`
## [1] 1086573856
##
## $`0.0436`
## [1] 2038806744
##
## $`0.049`
## [1] 2174903540
```

Because the plot is of a spectral distribution, and the y axis expresses how much of the variance in the time series a given frequency explains, mapping power to cosine amplitude is simple. Power, or height, at a given frequency is $2\sigma^2_\omega$ (multiplied by two because the we scan over $\omega = [-1/2, 1/2]$ picking up variance). The amplitude of each cosine wave directly corresponds to how much variance it creates, so we'll set amplitude equal to $\sqrt{(}2)\sqrt{(}power_\omega)$ for each cosine wave.

All that remains is build the corresponding cosine waves and stitch them together into a single waveform.

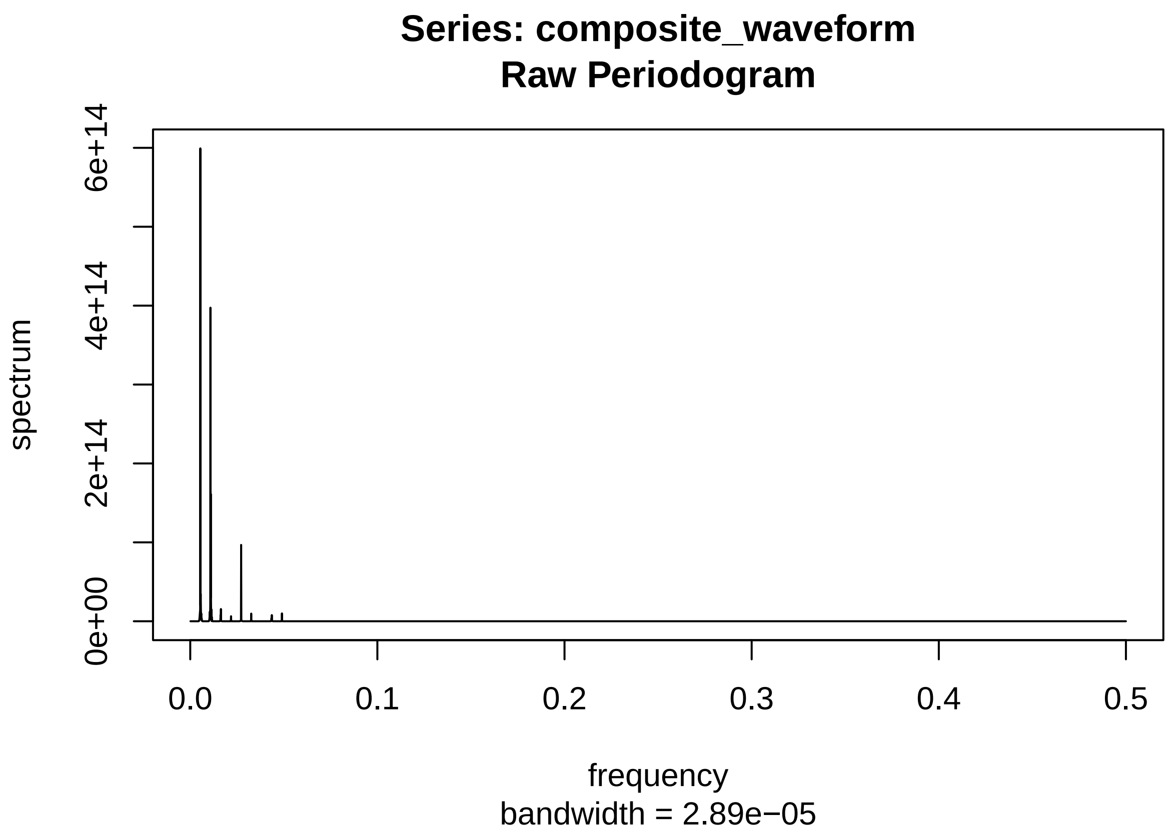


Fake Sax Composite Waveform
Amplitude
2e+06
1e+06
−1e+06
0
500
1000
1500
2000
Time




It doesn't seem to look like the sax time series, but some of that is due to how consistent it is. That's a given, because my pitch is more inconsistent than that of the computer generated sound wave. That said, there does seem to be a wider range of frequencies represented than in the keyboard periodogram, which was one of the things we were trying to reproduce. Let's check the periodogram for the composite waveform.

```
spec.pgram(composite_waveform, log = "no")
```




We seem to have reproduced the spectral distribution very well! This looks exactly like the periodogram of the T = 5,000 sample of the sax series seen above. Regardless, the best test will be done by ear. I've generated a longer version of this time series and downloaded it as an audio file titled "finalfakesax.wav".

It's worth noting that some compression of the cosine waves had to be done in order to download it, and the pitch being right is somewhat subject to picking the right sampling rate.

However, the pitch sounds roughly right on average, and the tone sounds a little richer than that of the keyboard note. I do think I can hear the accentuated harmonics. There is also an extreme vibrato, which makes it sound not great. It is visible in the time series plot for the fake saxophone note–there is some very strong low frequency structure that you don't really see in the sax time series. I'm not totally sure where that comes from given the similarity between the periodograms for the sax and composite waveform, and how apparent it is in one compared to the other. I'm inclined to think that's there because of a sampling issue or some problem with the generation and concatenation of the cosines.

If I were to continue, my next steps would be to generalize this process. Ideally, we could enter a keyboard note, find the fundamental frequency, and reproduce a new time series with the harmonic series of a saxophone building off of that given fundamental.