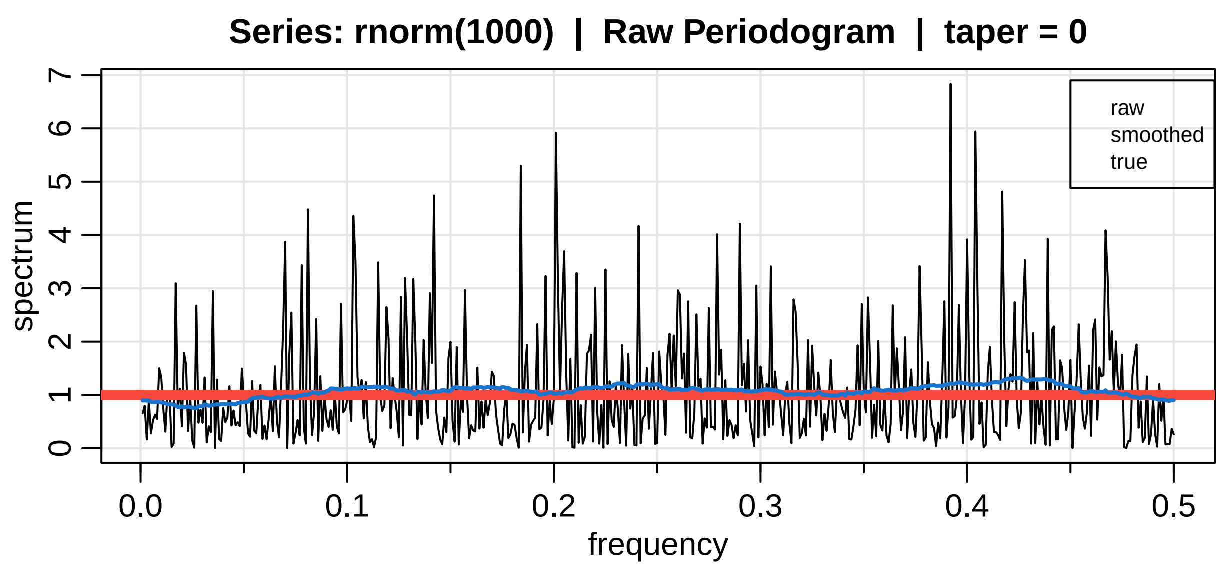
Spectral Analysis, additional examples

In this lecture, we'll cover various examples regarding spectral analysis.

One of the important technique while reading the periodogram is "smoothing".

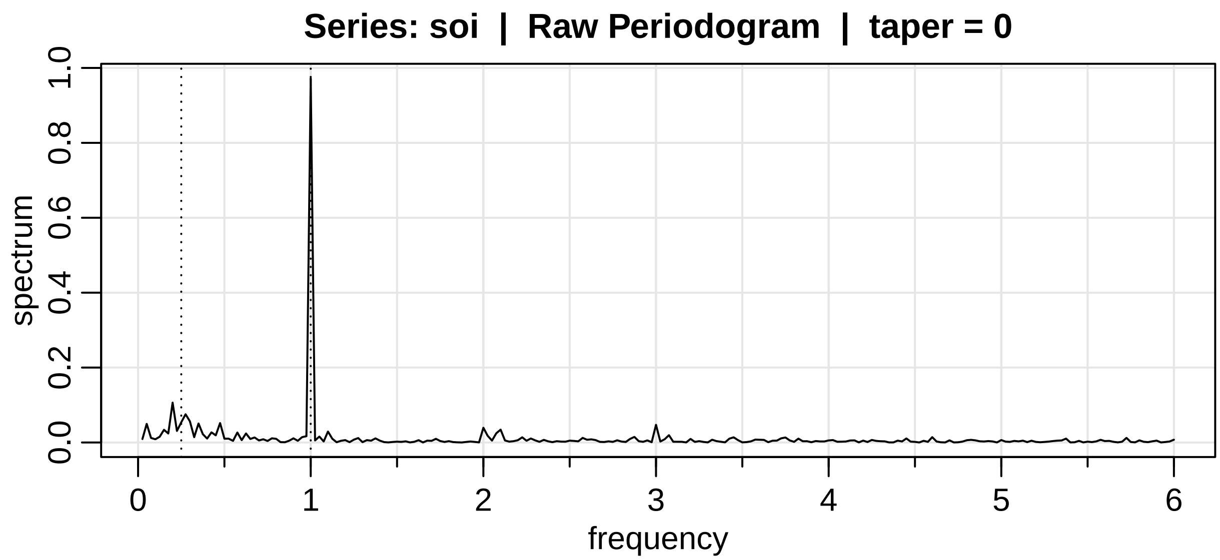
```
set.seed(1)
u = mvspec(rnorm(1000), col=1) # periodogram
abline(h=1, col=2, lwd=5,lty=1) # true spectrum
lines(u$freq, filter(u$spec, filter=rep(1,101)/101, circular=TRUE), col=4, lwd=2) # add the smooth
legend(0.45,6.9,legend=c("raw","smoothed","true"),col=c(1,4,2),cex=0.7)
```



Based on our knowledge, theoretically, white noise should have a flat line when we're looking at a spectral density. But in reality, it's hard to see that when we look at the periodogram. To analyze spectral analysis, we often conduct smoothing (one of nonparametric analysis) to the periodogram.

Let's start with `soi` series from `astsa` package.

```
soi.per=mvspec(soi) #raw periodogram
abline(v=1/4, lty="dotted")
abline(v=1, lty="dotted")
```



In this document, we're going to use `mvspec()` function from the `astsa` package.

You can observe a pick at frequency=1 (1 year). You can also see small pick at around 1/4, but it's hard to recognize. However, you can still read the amplitude:

```
soi.per$details[10,] #frequency=0.25 (4 year cycle)
```

```
## frequency    period  spectrum
##    0.2500    4.0000    0.0537
```

```
soi.per$details[40,] #frequency=1 (1 year cycle)
```

```
## frequency    period  spectrum
##    1.0000    1.0000    0.9722
```

```
length(soi)
```

```
## [1] 453
```

```
soi.per$n.used #nearest integer (bigger than n), multiple of 2,3,5
```

```
## [1] 480
```

While length of the `soi` series is 453, it's actually using another n=480. It's because we add 'padding' to the data, so that the length of the data will be multiples of 2,3, 5 (so that we can easily compute the fourier frequencies). R will automatically add 0's to the end of the time series (padding).

For smoothing, we're introducing frequency band, $\mathcal{B}$.

For a frequency of interest ω, we have a band, L fourier frequencies close to ω.

- $\omega_j = j/n$.
- Band is given as collection of the frequencies of the form $\omega^* = \omega_j + k/n$,

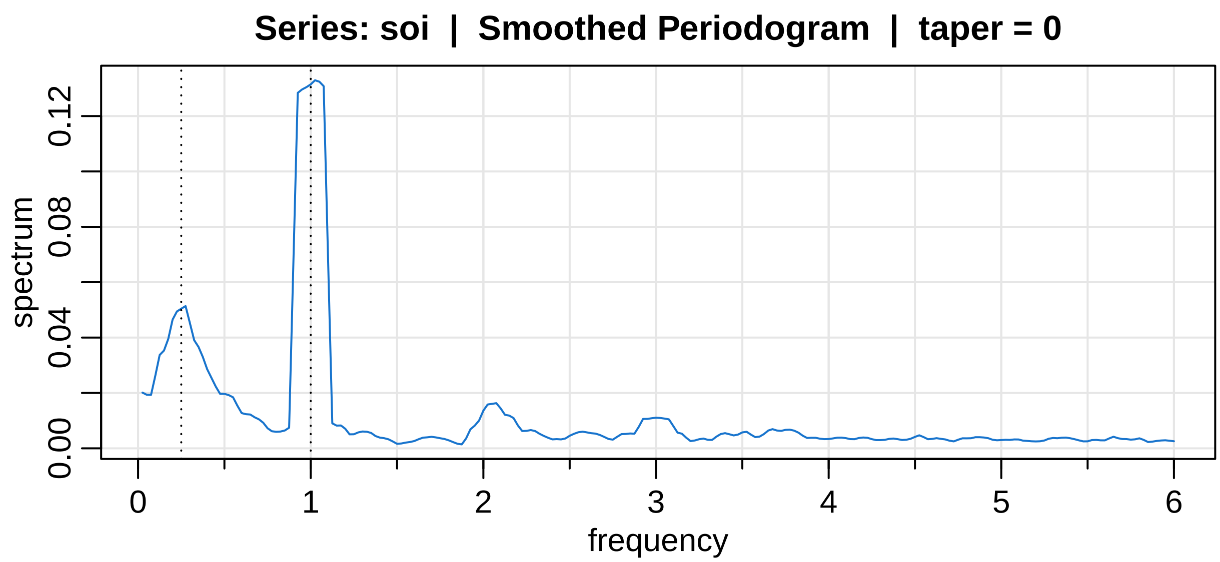
$$\mathcal{B} = \left\{\omega^* : \omega_j - \frac{m}{n} \leq \omega^* \leq \omega_j + \frac{m}{n}\right\}$$

Smoothed periodogram is given as the average of the periodogram values:

$$\bar{f}(\omega) = \frac{1}{2m+1} \sum_{k=-m}^{m} I(\omega_j + k/n),$$

Let's choose span=2m+1=9 for the example.

```
soi.per.span9=mvspec(soi,span=9,col=4)
abline(v=1/4, lty="dotted")
abline(v=1, lty="dotted")
```




```
soi.per.span9$details[10,] #frequency=0.25 (4 year cycle)
```

```
## frequency    period  spectrum
##    0.2500    4.0000    0.0504
```

```
soi.per.span9$details[40,] #frequency=1 (1 year cycle)
```

```
## frequency    period  spectrum
##    1.0000    1.0000    0.1315
```

```
soi.per.span9$n.used
```

```
## [1] 480
```

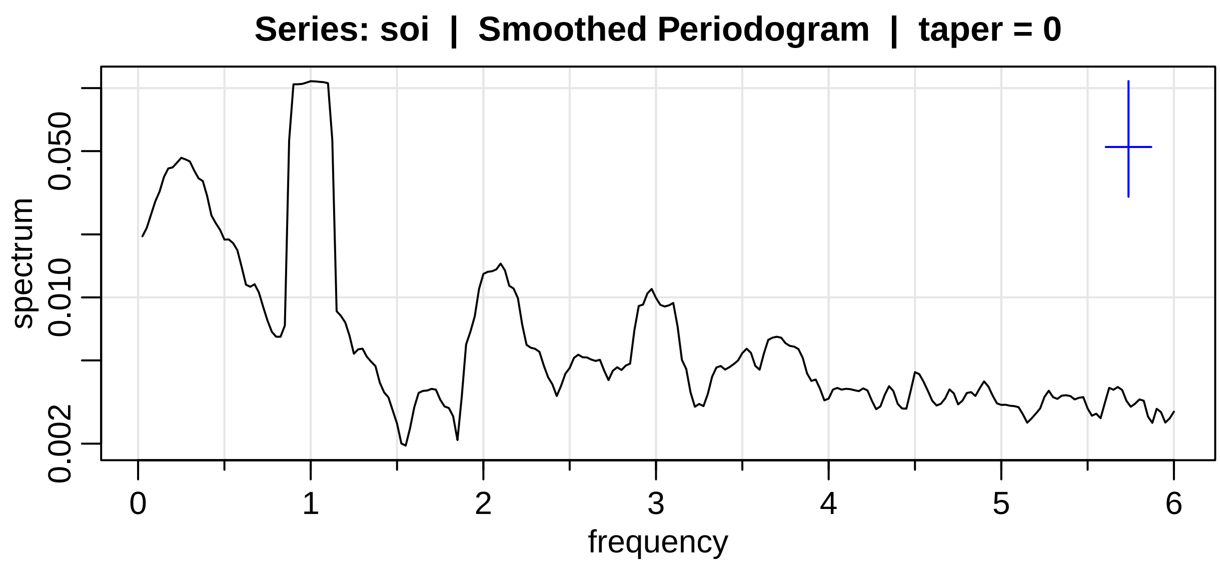
```r
soi.per.span9$df
```

```
## [1] 16.10667
```

After smoothing,

$$\frac{df * \bar{f}(\omega)}{f(\omega)} \sim \chi^2_{df}$$

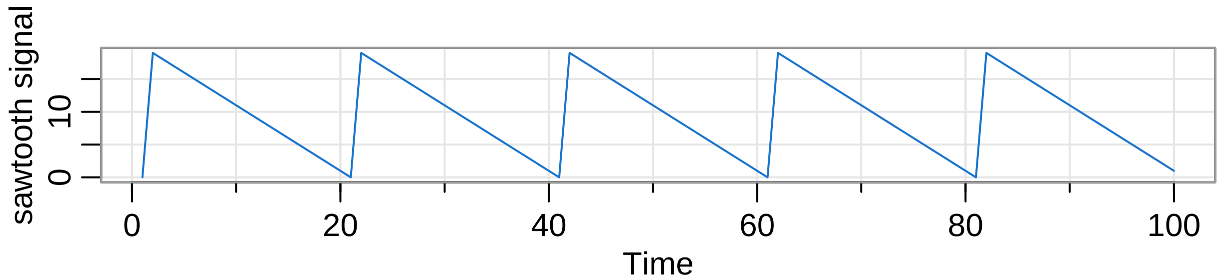
```r
soi.per.log=mvspec(soi,spans = 11,log = 'yes')
```

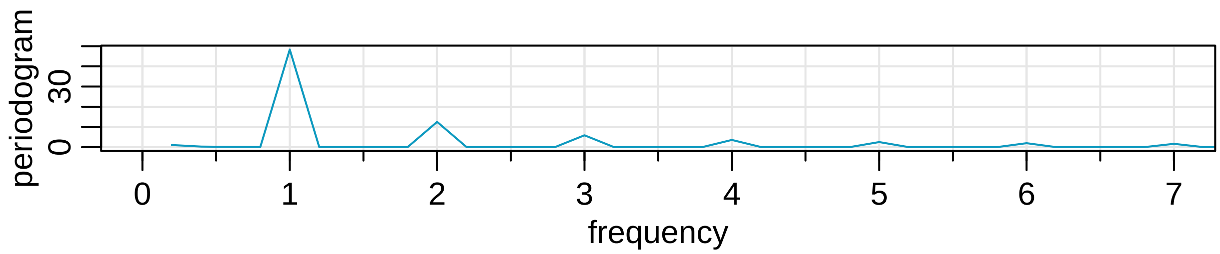


This is log scale periodogram, with confidence interval (vertical part of cross mark at the top right) and the span length (horizontal part of the cross mark).

Another thing you can notice here is that there are small peaks at frequency 2, 3, and 4. This is because of "Harmonics".

```r
y = ts(rev(1:100 %% 20), freq=20) # sawtooth signal
par(mfrow=2:1)
tsplot(1:100, y, ylab="sawtooth signal", col=4)
mvspec(y, main="", ylab="periodogram", col=rgb(.05,.6,.75), xlim=c(0,7))
```





In this simulated example, there is a pure sawtooth signal making one cycle every 20 points.

Theoretically, it should show only one peak at frequency=1.

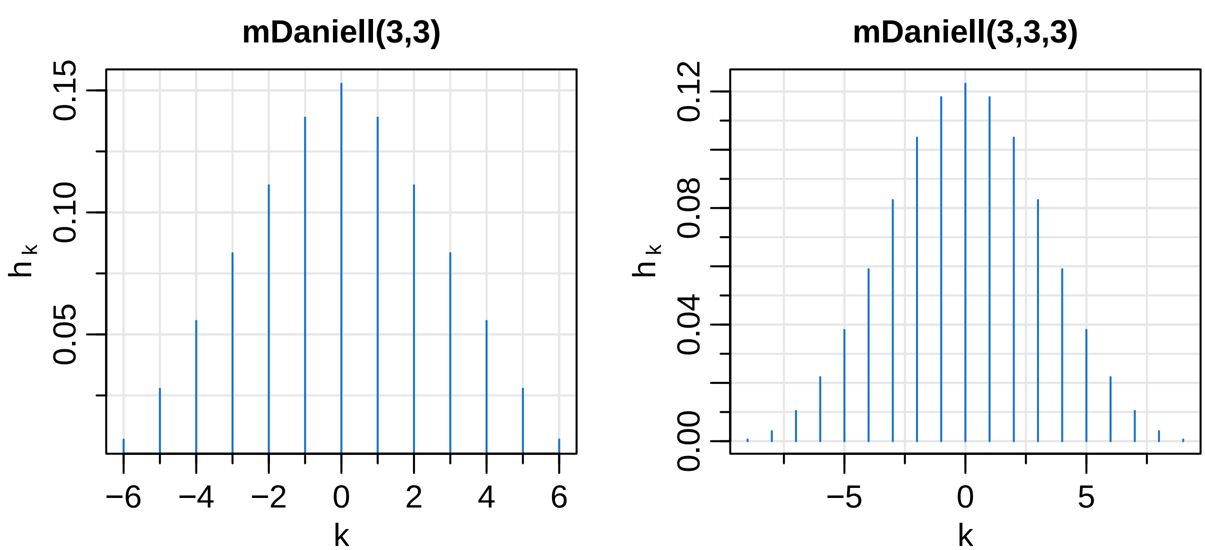
However, since signal is having a 'sawtooth' shape, not sinusoids (cosine-sine shape), it's showing additional peaks at frequency=2, 3, and so on. Therefore, if you see additional peaks at integer frequencies, it's maybe just because of harmonics, not because there's meaningful information there.

You can also choose different type of smoothing, not just averaging, but using different types of 'weights' to get weighted average.

```
(dm = kernel("modified.daniell", c(3,3))) # for a list
```

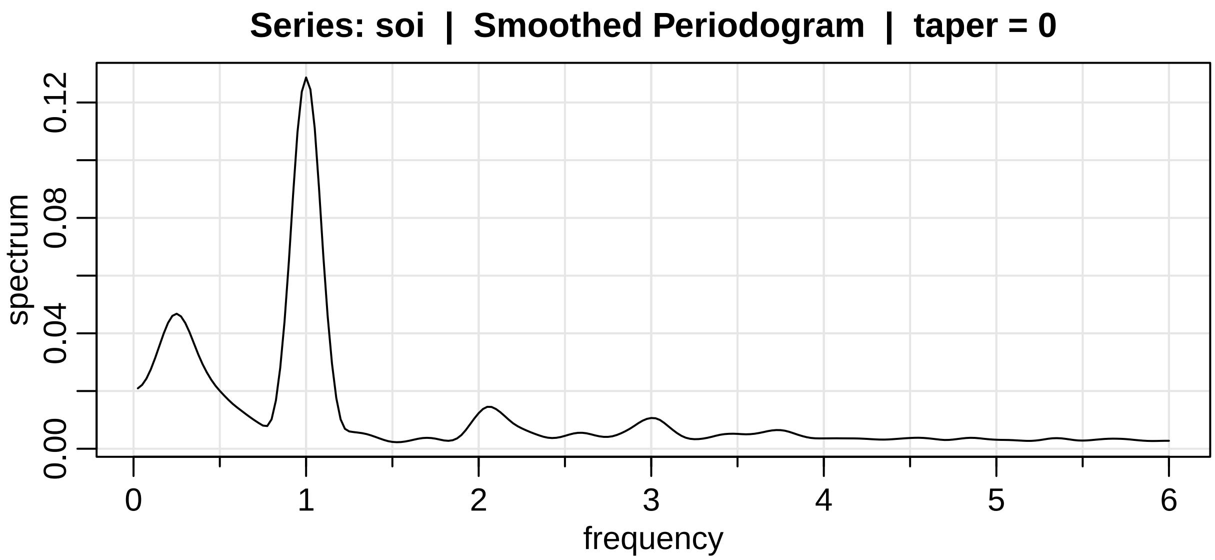
```
## mDaniell(3,3)
## coef[-6] = 0.006944
## coef[-5] = 0.027778
## coef[-4] = 0.055556
## coef[-3] = 0.083333
## coef[-2] = 0.111111
## coef[-1] = 0.138889
## coef[ 0] = 0.152778
## coef[ 1] = 0.138889
## coef[ 2] = 0.111111
## coef[ 3] = 0.083333
## coef[ 4] = 0.055556
## coef[ 5] = 0.027778
## coef[ 6] = 0.006944
```

```
# the figure with both kernels
par(mfrow=1:2, mar=c(3,3,2,1), mgp=c(1.6,.6,0))
plot(kernel("modified.daniell", c(3,3)), ylab=expression(h[~k]), cex.main=1, col=4,  panel.first=Grid()
plot(kernel("modified.daniell", c(3,3,3)), ylab=expression(h[~k]), cex.main=1, col=4,  panel.first=Grid
```



These are the shapes of modified Daniell kernel. The bar heights represent the weights for the nearby frequencies.

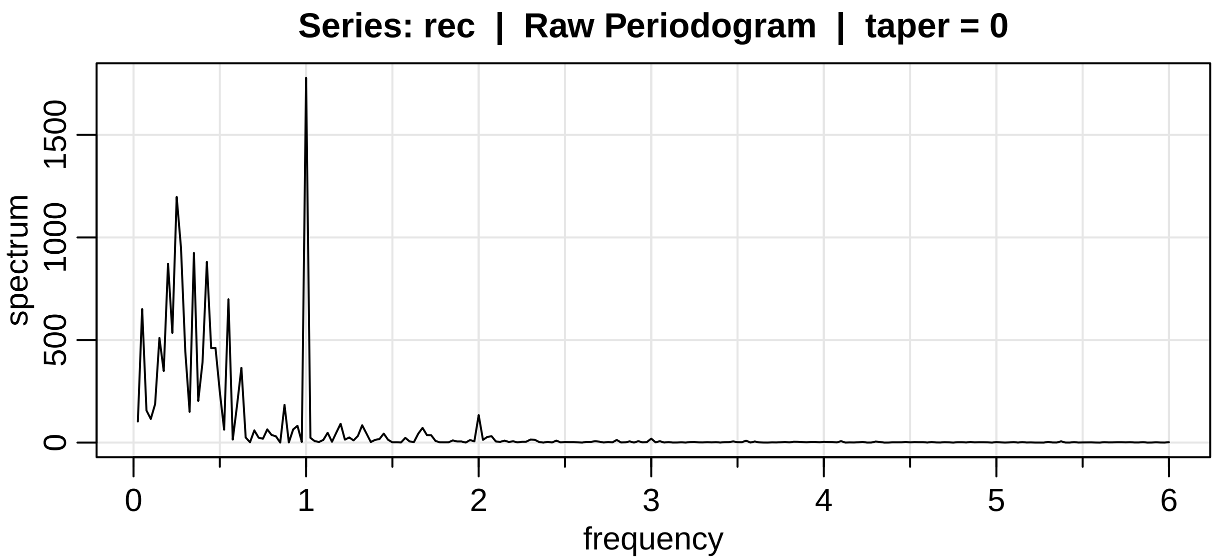
```
soi.daniell=mvspec(soi,kernel("modified.daniell", c(3,3,3)))
```



Series: soi | Smoothed Periodogram | taper = 0
spectrum
0.00
0.04
0.08
0.12
0
1
2
3
4
5
6
frequency

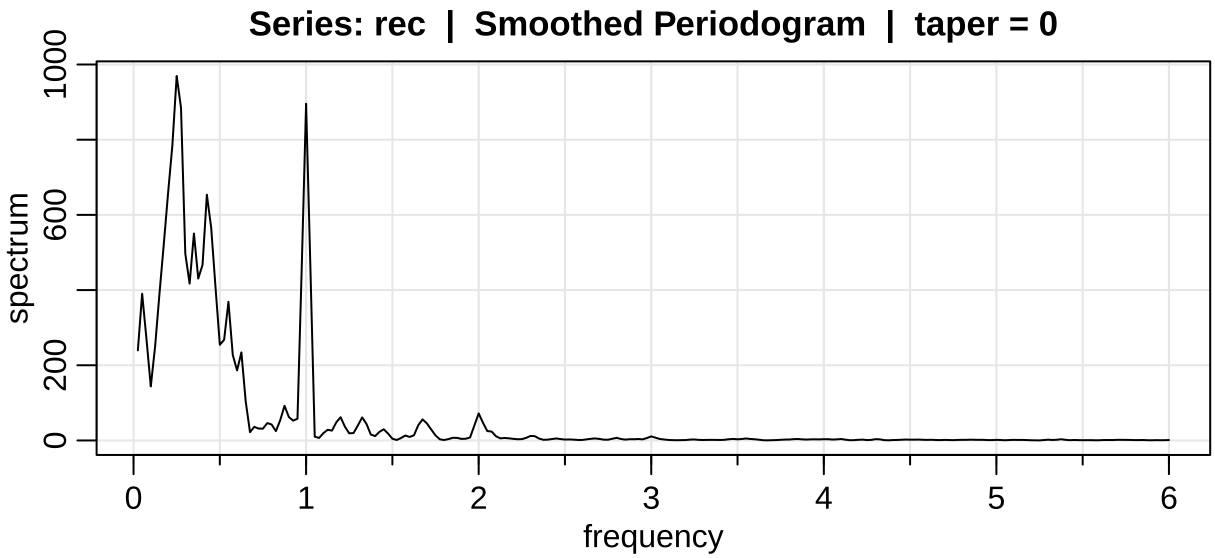
Now, let's work with another example.

```
mvspec(rec)
```

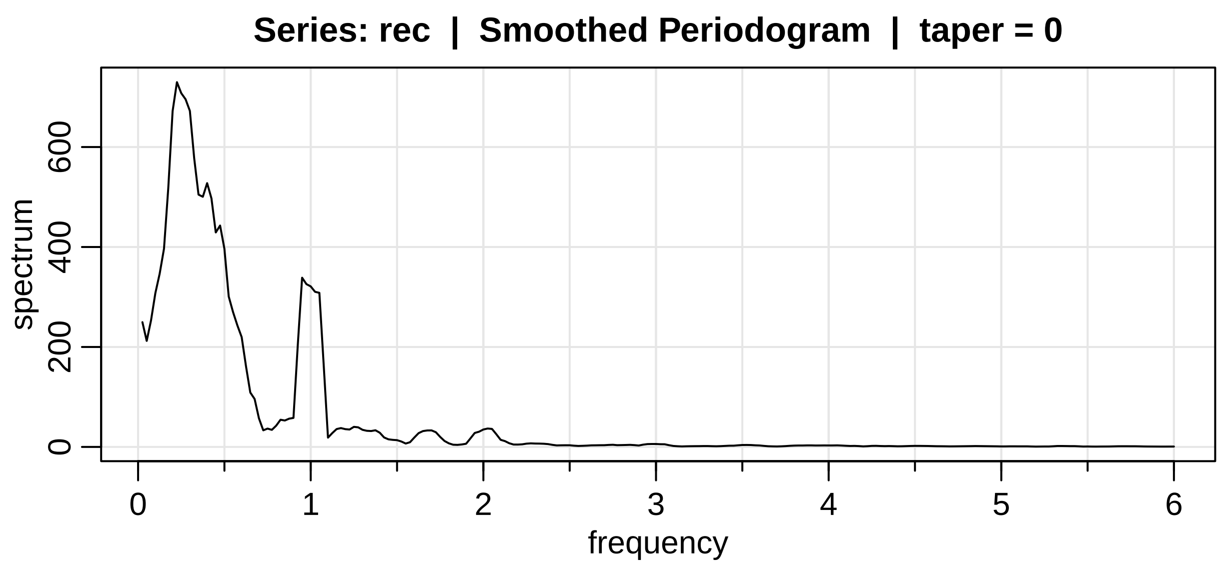


It seems like smoothing can also help here.

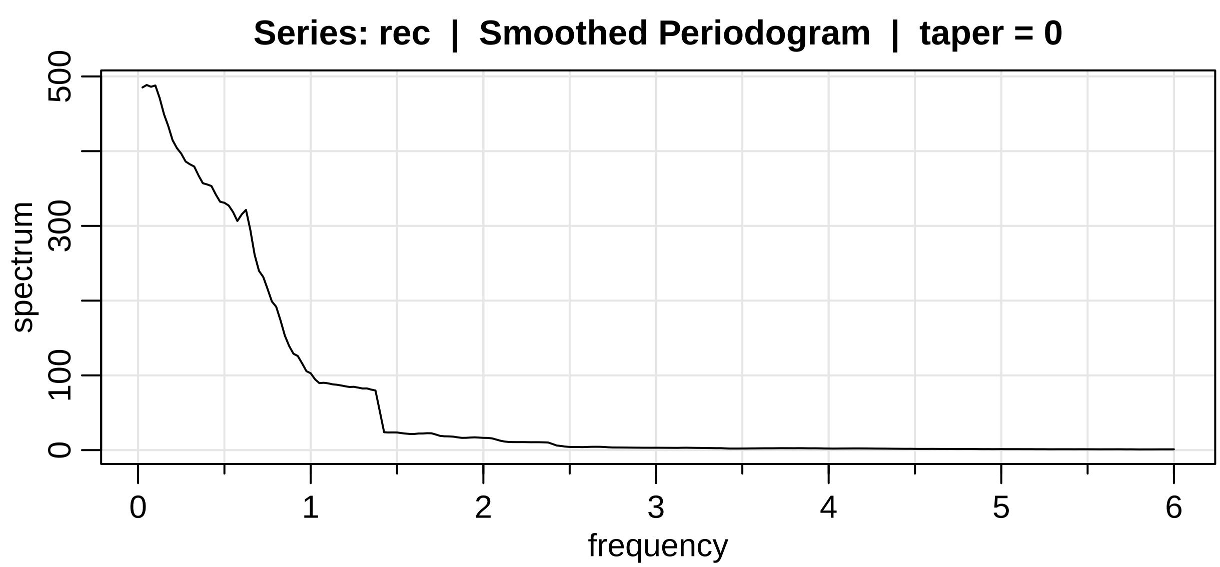
```
mvspec(rec,spans=3)
```



```r
mvspec(rec,spans=7)
```




```r
mvspec(rec,spans=33)
```



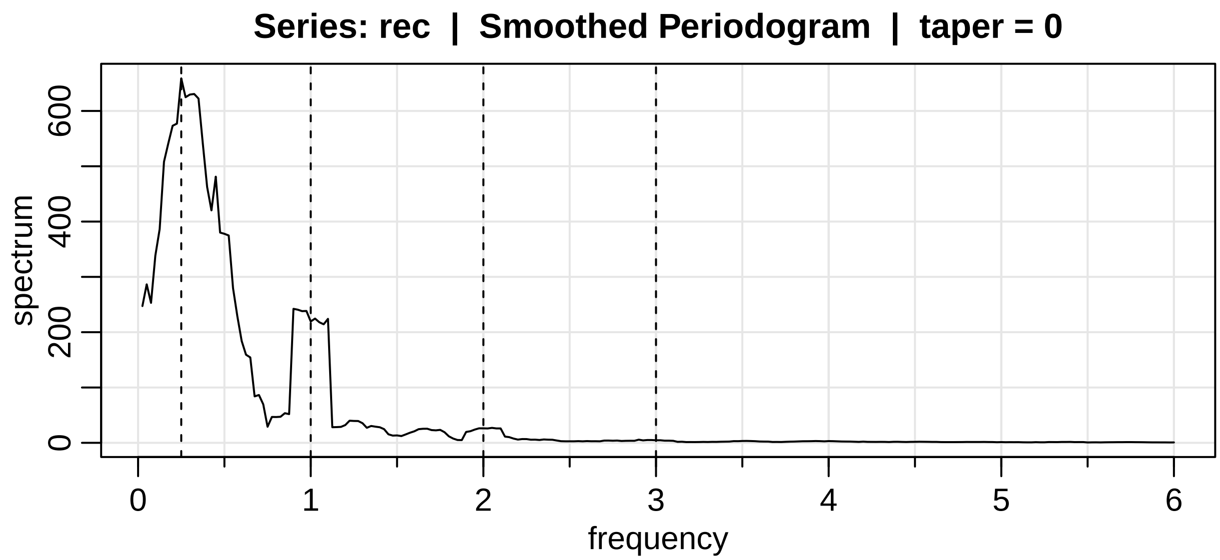
You can see that if we increase smoothing parameter too much, it's hard to recognize the peaks. It's related to the resolution.

Resolution is an ability of a spectrum estimate to represent fine structure in the frequency properties of the series, such as narrow peaks in the spectrum.

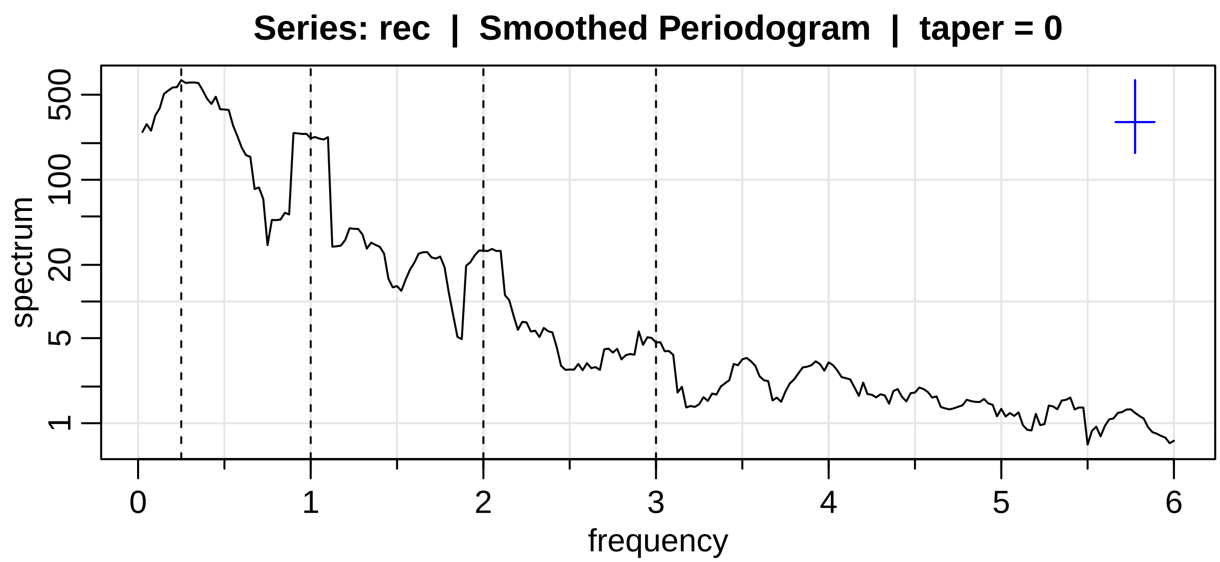
If the spectrum of a series contains two narrow peaks closer than the bandwidth, the resulting broad peaks in the spectrum estimation overlap and form a single pick: "failed" to "resolve" the peaks.

You may try different span parameters or different kernels to find appropriate smoothing parameters.

```
k=kernel('daniell',4)
rec.per=mvspec(rec,k)
abline(v=c(.25,1,2,3), lty=2)
```




```
rec.per=mvspec(rec,k,log='yes')
abline(v=c(.25,1,2,3), lty=2)
```




```
rec.per$df
```

```
## [1] 16.9875
```

```
rec.per$details[10,] #period=4 year
```

```
## frequency    period  spectrum
##    0.2500    4.0000  658.9607
```

```r
rec.per$details[40,] # period= 1 year
```

```
## frequency    period  spectrum
##    1.0000    1.0000  219.3919
```