STAT 443 Lab 11

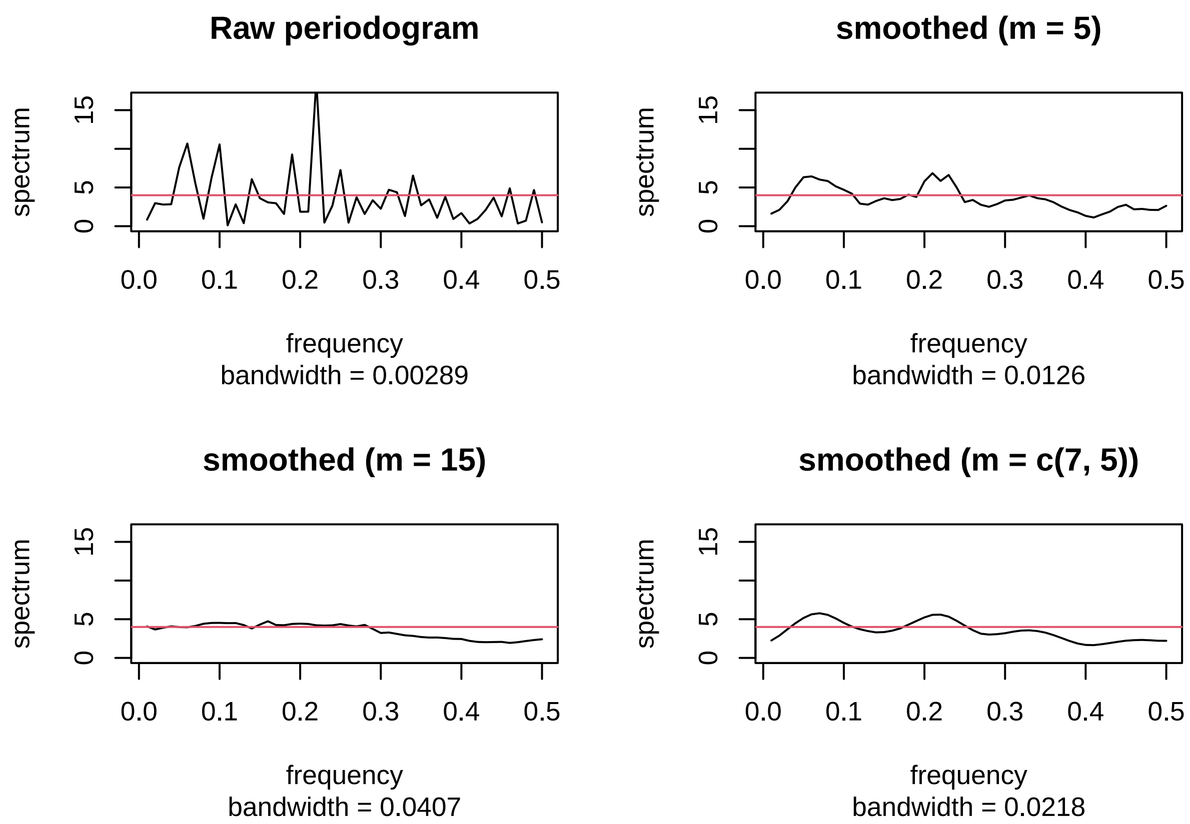
Derek Situ

2022-04-04

Question 1

1a

```r
set.seed(2022)
z <- arima.sim(list(), n = 100, sd = 2)

par(mfrow = c(2, 2))

spec.pgram(z, log = "no", ylim = c(0, 16.6), main = "Raw periodogram")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(5),
           main = "smoothed (m = 5)")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(15),
           main = "smoothed (m = 15)")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(7, 5),
           main = "smoothed (m = c(7, 5))")
abline(h = 4, col = 2)
```

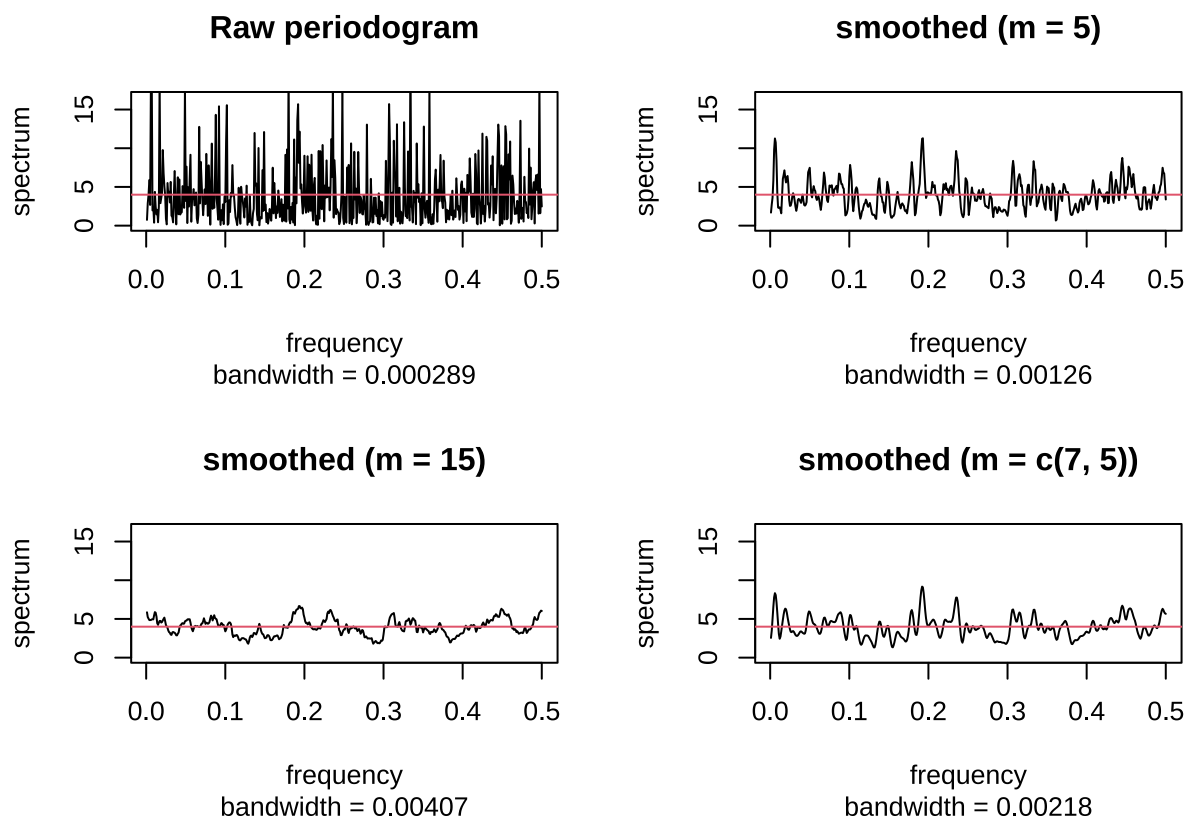


The spectrum is represented by the red line which is constant at $\frac{4}{\pi}$, noting that the y-axis shows values of the periodogram/spectrum multiplied by π. The raw periodogram looks quite different than the spectrum, with more spikes and dips. Smoothing with `spans = c(5)` or `spans = c(7, 5)` result in smoother periodograms that look more like the spectrum but still show the largest, most dominant frequencies in the sample. Smoothing with `spans = c(15)` smoothes the periodogram so much that it is almost constant like the spectrum but this does not reveal much information about the sample we generated.

1b

```r
set.seed(2022)
z <- arima.sim(list(), n = 1000, sd = 2)

par(mfrow = c(2, 2))

spec.pgram(z, log = "no", ylim = c(0, 16.6), main = "Raw periodogram")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(5),
           main = "smoothed (m = 5)")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(15),
           main = "smoothed (m = 15)")
abline(h = 4, col = 2)
```

```
spec.pgram(z, log = "no", ylim = c(0, 16.6), spans = c(7, 5),
           main = "smoothed (m = c(7, 5))")
abline(h = 4, col = 2)
```

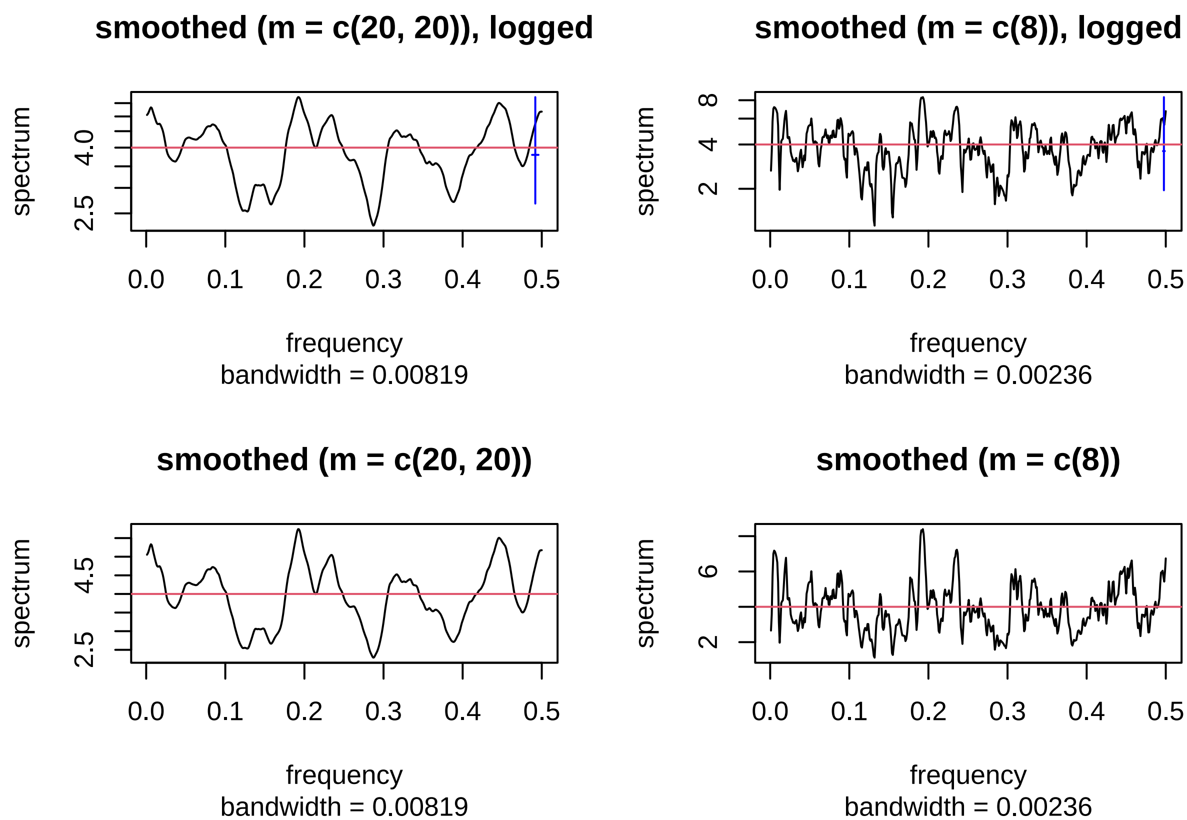


After increasing the sample size we can see that there is more variation in all periodograms compared to the $n = 100$ case. Smoothing with `spans = c(15)` now seems to be the most appropriate choice, as the periodogram looks most like the spectrum, and still shows the most important frequencies in the sample.

1c

```
par(mfrow = c(2, 2))

spec.pgram(z, spans = c(20, 20),
           main = "smoothed (m = c(20, 20)), logged")
abline(h = 4, col = 2)

spec.pgram(z, spans = c(8),
           main = "smoothed (m = c(8)), logged")
abline(h = 4, col = 2)

spec.pgram(z, log = "no", spans = c(20, 20),
           main = "smoothed (m = c(20, 20))")
abline(h = 4, col = 2)
```

```
spec.pgram(z, log = "no", spans = c(8),
           main = "smoothed (m = c(8))")
abline(h = 4, col = 2)
```



We see above that logging makes no obvious difference in the shape of the periodogram. I also compared smoothing with `spans = c(20, 20)` and `spans = c(8)` and saw that the first choice (logged or not logged), shown by the periodograms on the left side, is better. It is more smooth than the other case, yet still shows the most important frequencies.

1d

If we have

$$\frac{2I(\omega)}{f(\omega)} \sim \chi_2^2,$$

then taking logs gives

$$log(2) + log(I(\omega)) - log(f(\omega)) \sim log\chi_2^2.$$

Now we see that the error, plus a small constant, has a log chi-squared distribution, or

$$log(I(\omega)) - log(f(\omega)) = v - log(2)$$

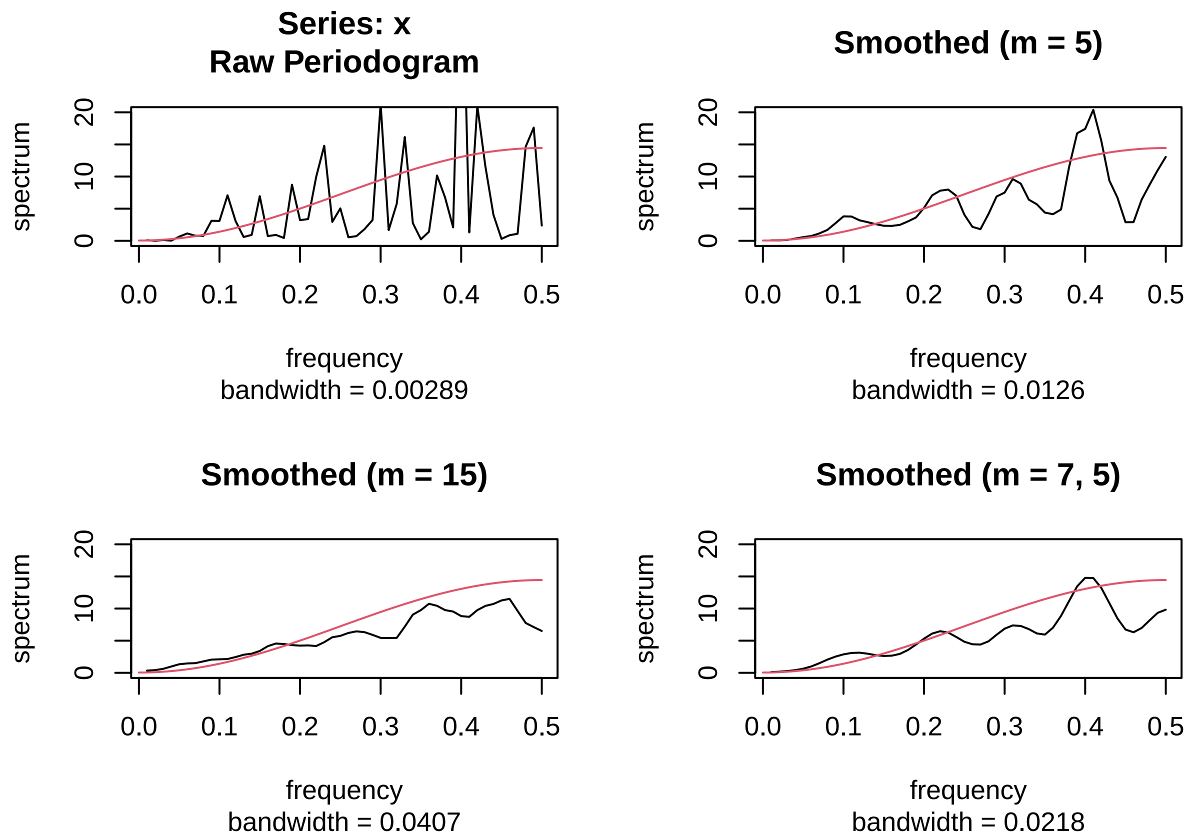
where $v \sim log\chi_2^2$.

This is easier to interpret that the original relationship which said that the ratio of twice the periodogram to the spectrum is chi-squared distributed. The result is consistent with (1c).

Question 2

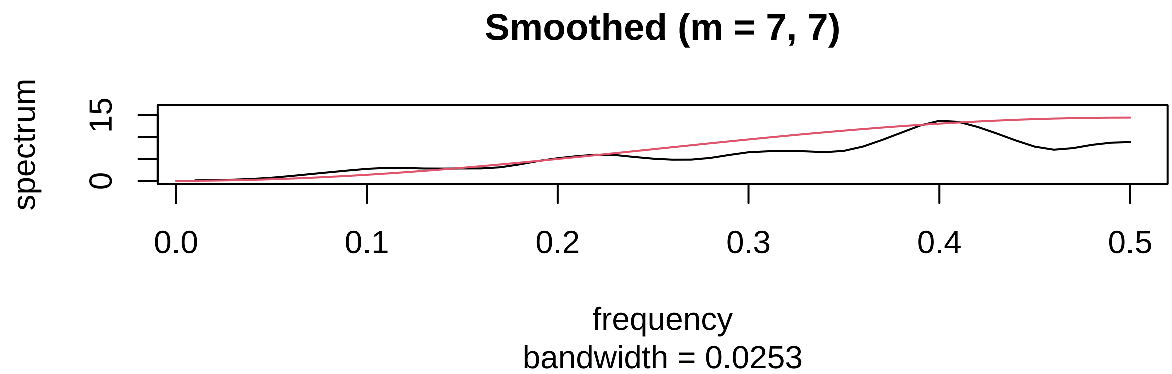
2a

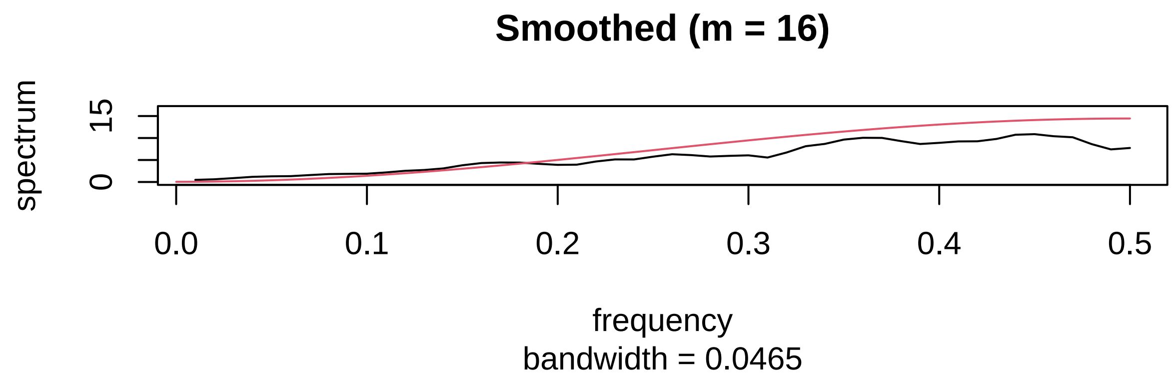
```
omega <- seq(0, pi, by = 0.01 * pi)
spectrum <- 7.24/pi * (1 - 1.8*cos(omega)/1.81)

set.seed(13)
x <- arima.sim(list(ma = c(-0.9)), n = 100, sd = 2)

par(mfrow = c(2, 2))

spec.pgram(x, log = "no", ylim = c(0, 20))
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(x, log = "no", ylim = c(0, 20),
           spans = c(5), main = "Smoothed (m = 5)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(x, log = "no", ylim = c(0, 20),
           spans = c(15), main = "Smoothed (m = 15)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(x, log = "no", ylim = c(0, 20),
           spans = c(7, 5), main = "Smoothed (m = 7, 5)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
```



We see that the raw periodogram has the most variation in the form of spikes and dips, and there is less variation as we smooth more. In particular, smoothing with `spans = c(7, 5)` or `spans = c(15)` gives periodograms that are somewhat similar to the spectrum, which is plotted in red.

```
par(mfrow = c(2, 1))

spec.pgram(x, log = "no", ylim = c(0, 16.6),
           spans = c(7, 7), main = "Smoothed (m = 7, 7)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(x, log = "no", ylim = c(0, 16.6),
           spans = c(16), main = "Smoothed (m = 16)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
```



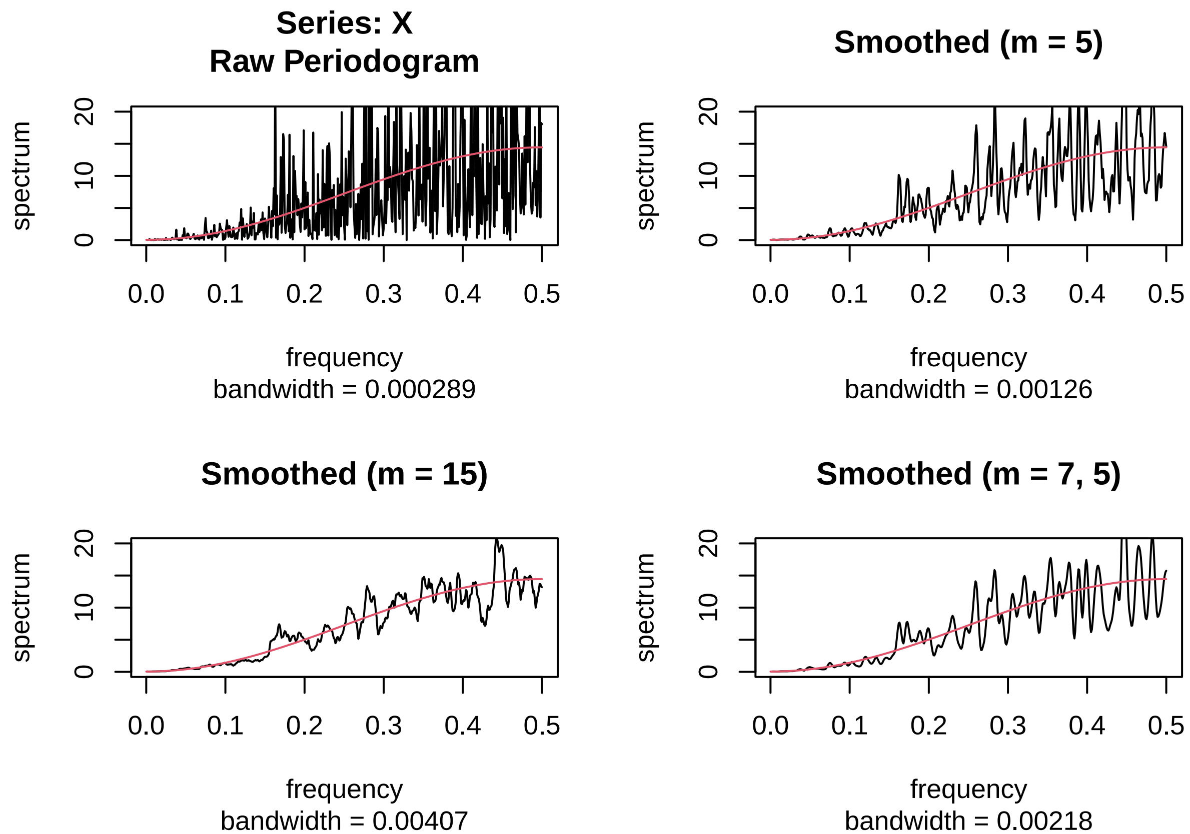


Smoothing with `spans = c(7, 7)` or `spans = c(16)` give periodograms that are pretty close to the spectrum, although at higher frequencies, the estimate is consistently under the spectrum.

2c

```
X <- arima.sim(list(ma = c(-0.9)), n = 1000, sd = 2)

par(mfrow = c(2, 2))

spec.pgram(X, log = "no", ylim = c(0, 20))
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(X, log = "no", ylim = c(0, 20),
           spans = c(5), main = "Smoothed (m = 5)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(X, log = "no", ylim = c(0, 20),
           spans = c(15), main = "Smoothed (m = 15)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(X, log = "no", ylim = c(0, 20),
```

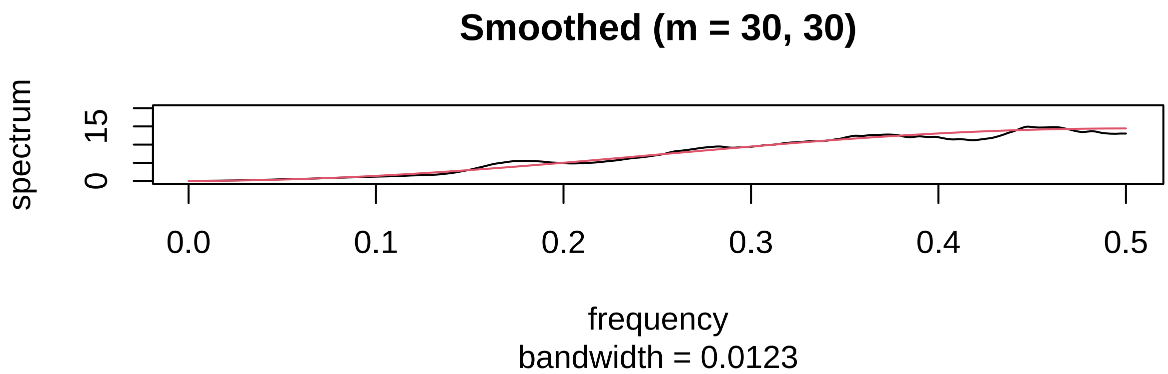
```
            spans = c(7, 5), main = "Smoothed (m = 7, 5)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
```

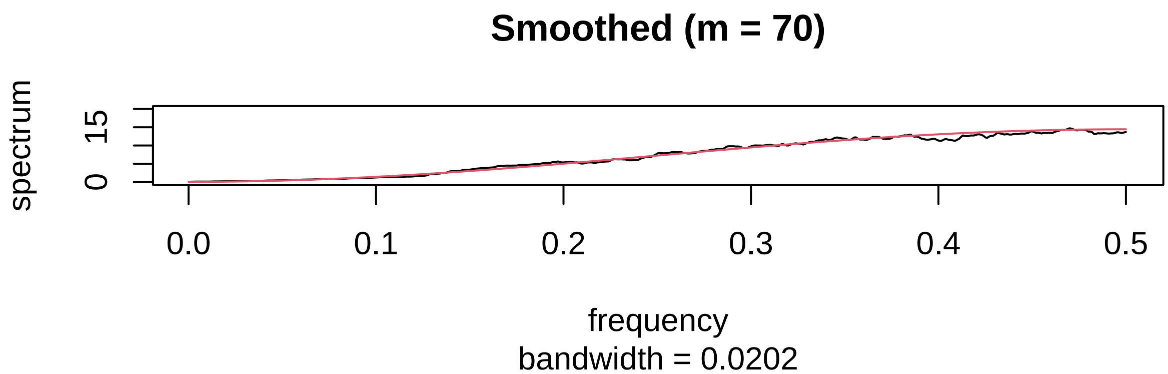


Since we have increased the sample size, there is more variation in the periodograms, especially when we smooth less. Even the smoothing parameters that led to quite good results in (2a) and (2b) lead to undersmoothed periodograms this time.

2d

```
par(mfrow = c(2, 1))

spec.pgram(X, log = "no", ylim = c(0, 20),
            spans = c(30, 30), main = "Smoothed (m = 30, 30)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
spec.pgram(X, log = "no", ylim = c(0, 20),
            spans = c(70), main = "Smoothed (m = 70)")
lines(x = omega / (2 * pi),
      y = spectrum * pi,
      type = "l", col = 2)
```





Smoothing with `spans = c(30, 30)` or `spans = c(70)` gives periodograms that are very close to the spectrum. As we can see, we needed more smoothing for a larger sample size, but the estimate performs better with a larger sample size, especially at higher frequencies.