Lab 11: Modifying the Periodogram

STAT 443

Caden Hewlett

2024-04-05

For the following, we $\{Z_t\}_{t\in\mathbb{Z}} \overset{\text{iid}}{\sim} \text{WN}(0, \sigma = 2)$.

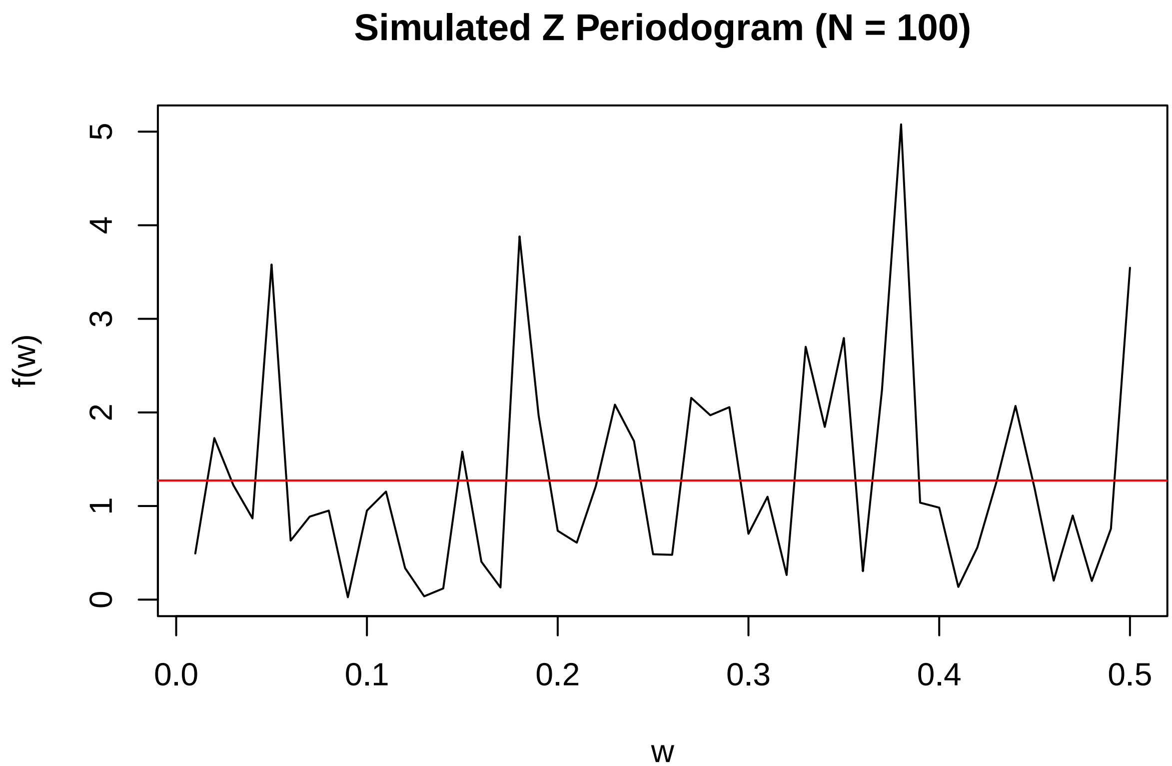
Question 1

Here we will compare the spectral density of white noise with smoothed periodograms obtained from simulated white noise samples.

Part A

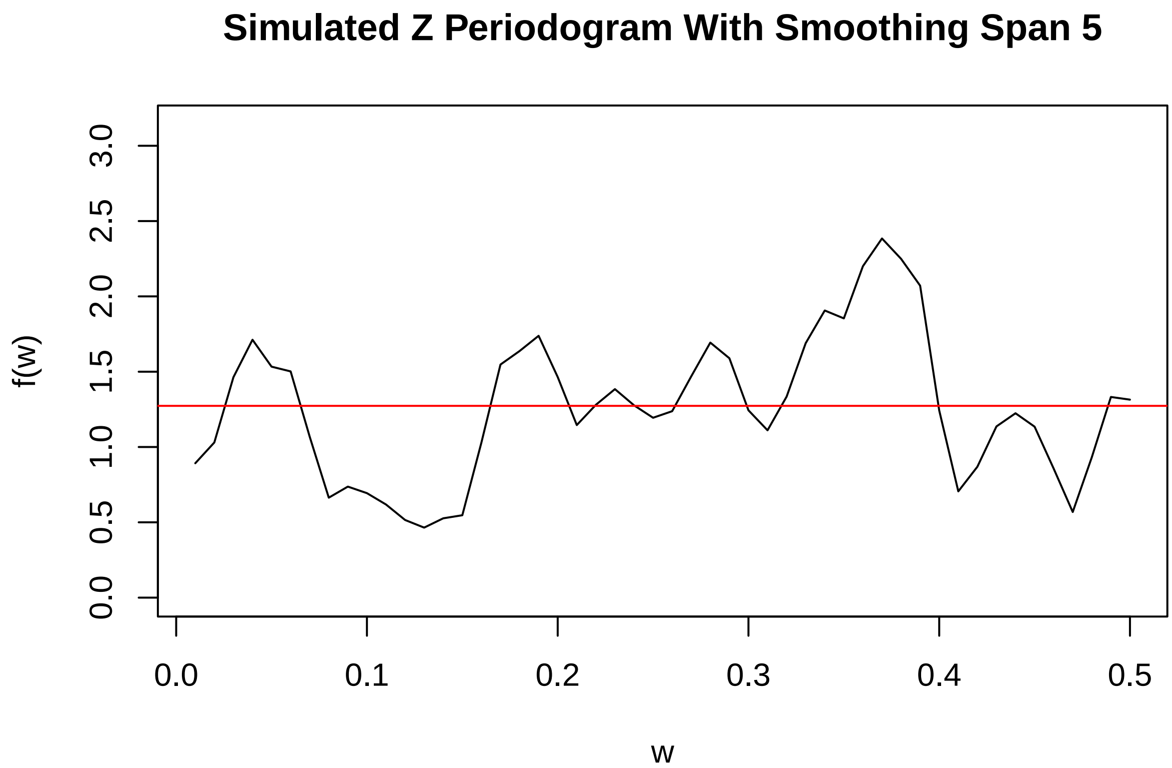
Use the `arima.sim` command, or otherwise, to simulate a series of length 100 from $\{Z_t\}_{t\in\mathbb{Z}}$. Use `spec.pgram` to create and plot the periodogram for your sample.

```
set.seed(1924)
WN = arima.sim(model = list(), n = 100, sd = 2)
f_w_WN = spec.pgram(WN, log = "no", plot = F)
plot(
  y = f_w_WN$spec / pi,
  x = f_w_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram (N = 100)"
)
abline(h = 4 / pi, col = 'red')
```

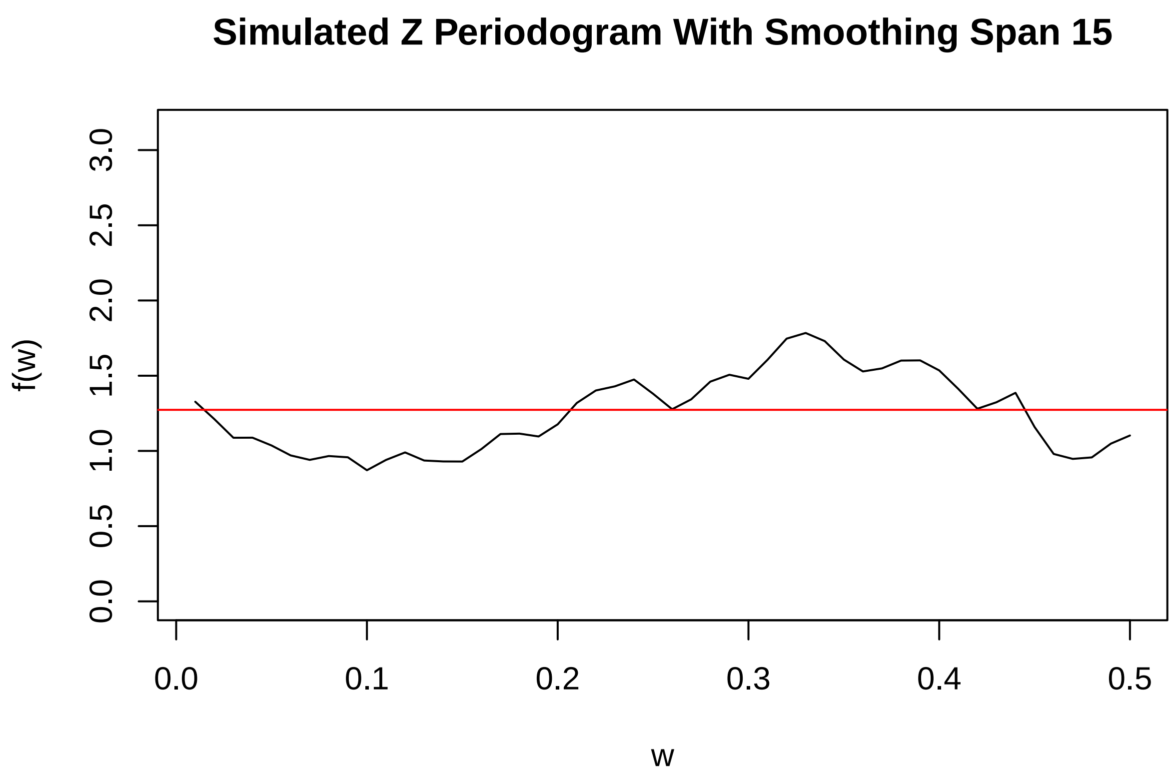


Now, smooth your periodogram by smoothing with spans (i) c(5), (ii) c(15), (iii) c(7,5).

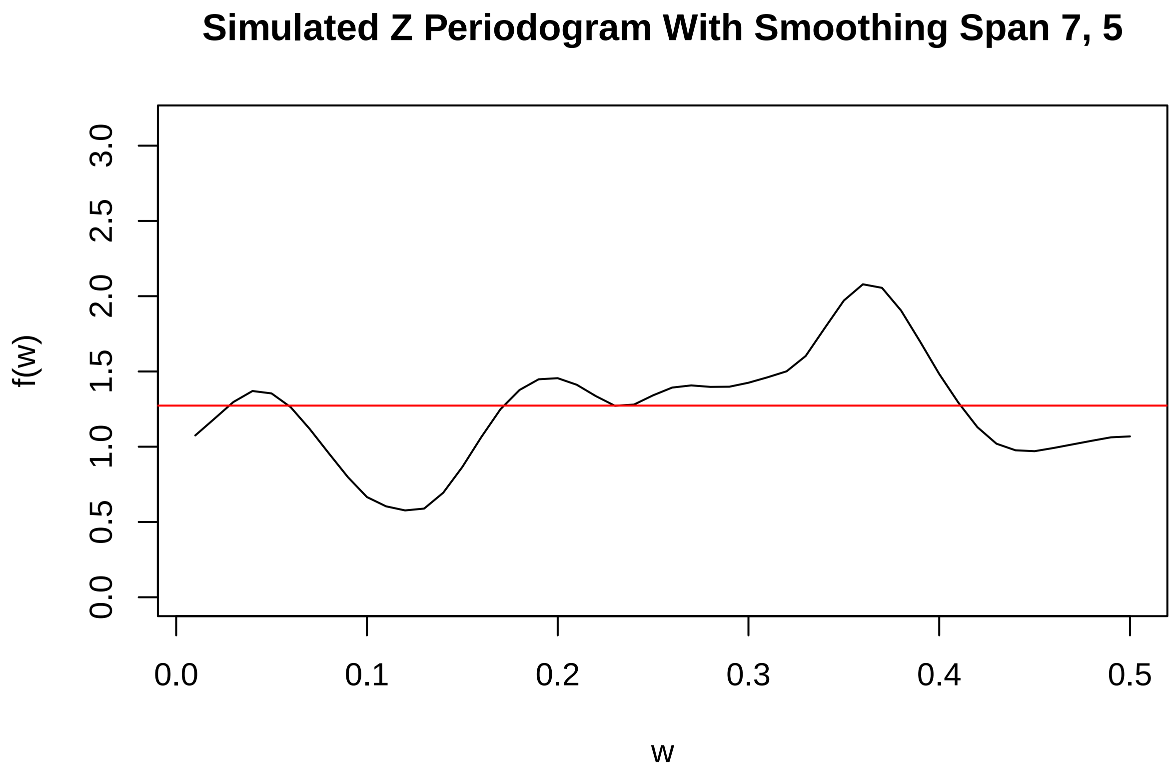
```
f_i_WN = spec.pgram(WN, log = "no", plot = F, spans = c(5))
plot(
  y = f_i_WN$spec / pi,
  x = f_i_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With Smoothing Span 5",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'red')
```




```r
f_ii_WN = spec.pgram(WN, log = "no", plot = F, spans = c(15))
plot(
  y = f_ii_WN$spec / pi,
  x = f_ii_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With Smoothing Span 15",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'red')
```




```
f_iii_WN = spec.pgram(WN, log = "no", plot = F, spans = c(7, 5))
plot(
  y = f_iii_WN$spec / pi,
  x = f_iii_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With Smoothing Span 7, 5",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'red')
```



Comments: Comment on what you observe, comparing the periodogram with the spectral density function.

Note that for a WN process, the theoretical spectral density is simply $\frac{\sigma^2}{\pi} = \frac{4}{\pi}$, which is indicated by the red lines on each plot. We note that in the original raw periodogram, the observed $I(\omega)$ values appear to roughly surround the true spectral density; however, there is a lot of noise. As we increase the one-dimensional smoothing span (as seen in plots **i** and **ii**), we see the smoothed periodogram begins to more closely surround the true spectrum, with decreased noise. Interestingly, the two-dimensional `c(7,5)` span is less close to the true spectrum than the `15` span, but appears to be a smoother curve overall. I would wager that the large `15` span is the most biased towards the true spectrum, while the `7, 5` span is the smoothest overall. The `5` span appears to be under-smoothed. One could argue that either the `15` or `7,5` spans are over-smoothed, depending on your target criteria. From a purely visual standpoint, the `15` span is closest to the "true" horizontal line.

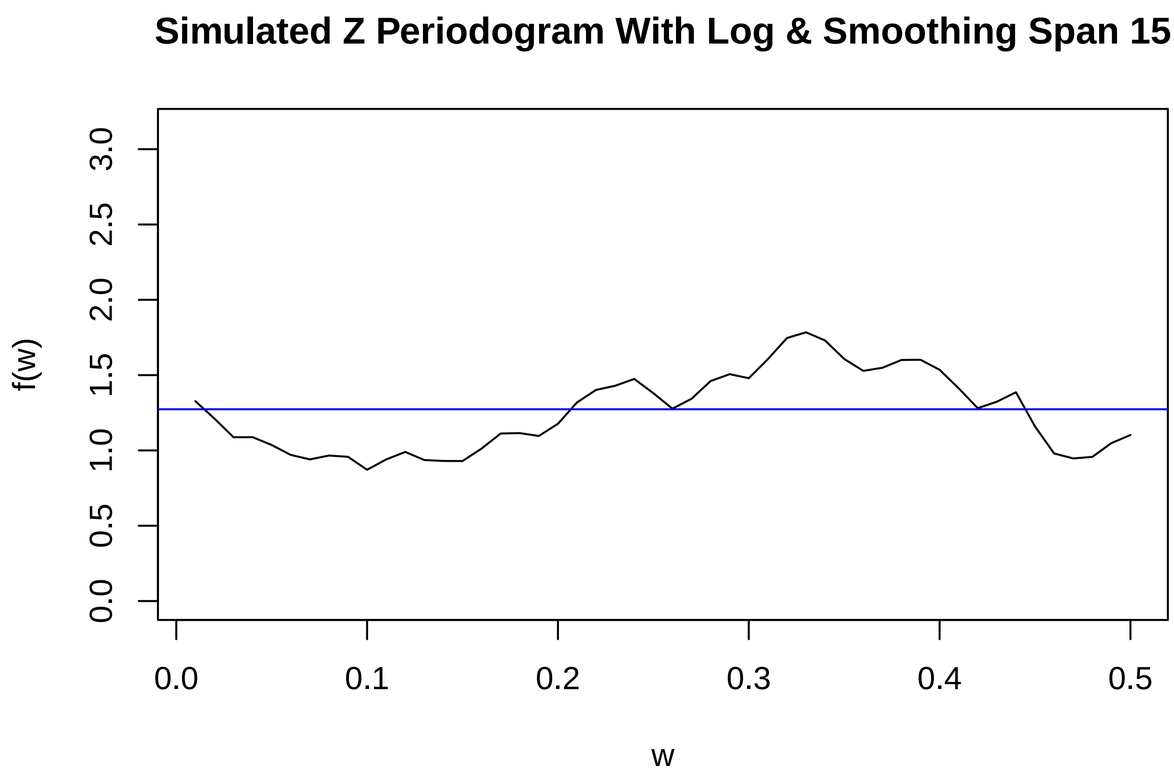
Part C

In terms of the shape at least, it should make no difference here whether we take logs of the periodogram of the data from (b). Play around with various choices of (i) taking logs or not and (ii) the spans argument, looking at no less than four special cases. Which choice was "best", and why?

Using Log, Span of 15

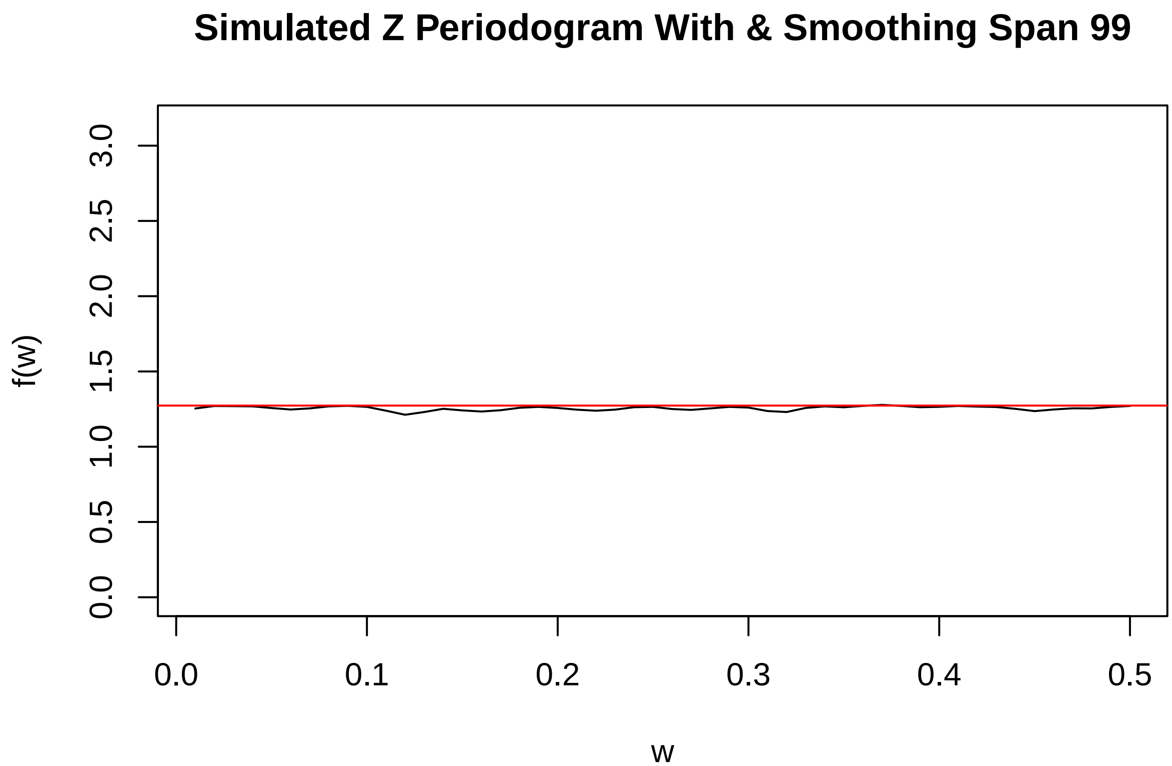
```
f_i_WN = spec.pgram(WN,  plot = F, spans = c(15))
plot(
  y = f_i_WN$spec / pi,
  x = f_i_WN$freq,
  type = 'l',
```

```r
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With Log & Smoothing Span 15",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'blue')
```



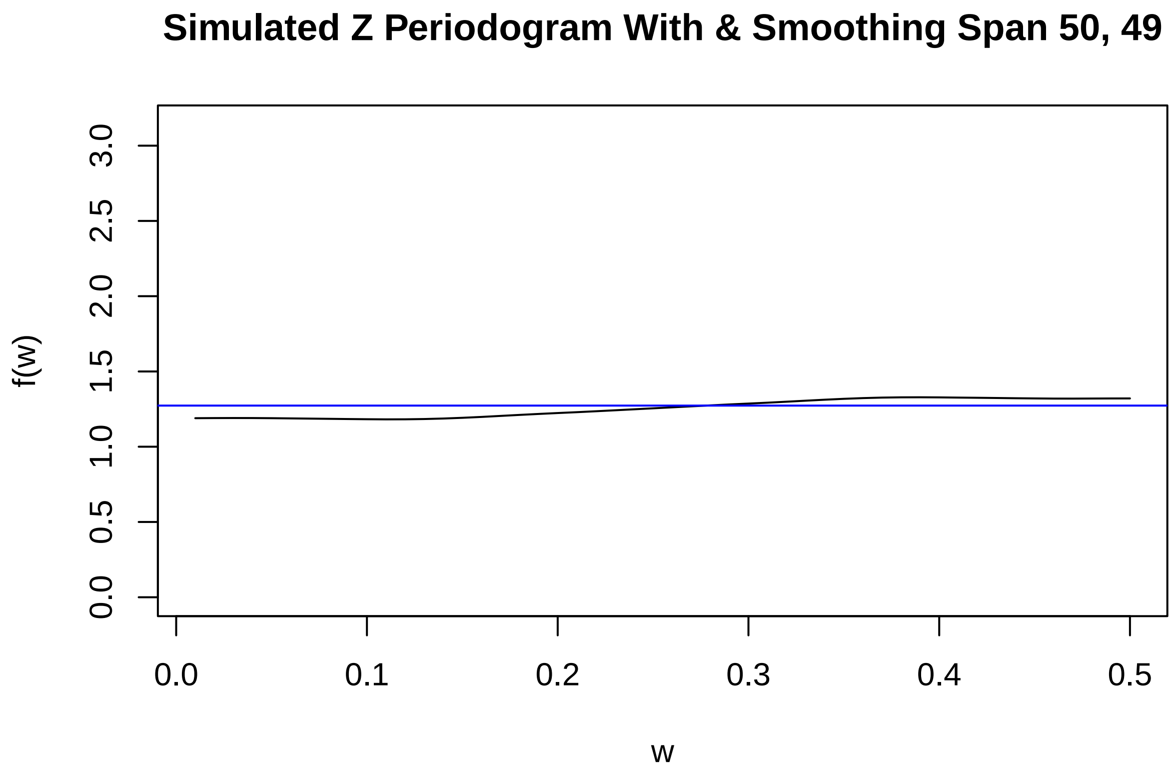
Using No Log, Span of 99

```r
f_i_WN = spec.pgram(WN, log = "no", plot = F, spans = c(99))
plot(
  y = f_i_WN$spec / pi,
  x = f_i_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With & Smoothing Span 99",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'red')
```



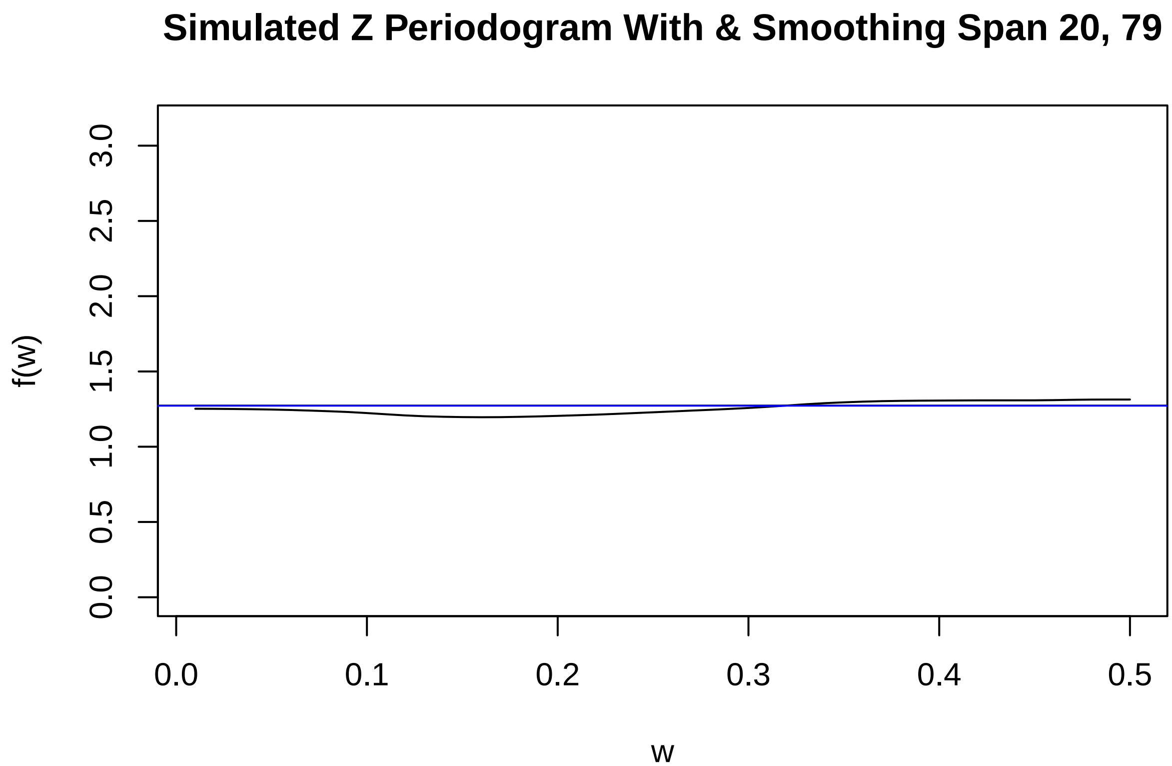
Using Log, Span of 50, 49

```
f_i_WN = spec.pgram(WN, plot = F, spans = c(50, 49))
plot(
  y = f_i_WN$spec / pi,
  x = f_i_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With & Smoothing Span 50, 49",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'blue')
```



Using Log, Span of 20, 79

```
f_i_WN = spec.pgram(WN, plot = F, spans = c(20, 79))
plot(
  y = f_i_WN$spec / pi,
  x = f_i_WN$freq,
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated Z Periodogram With & Smoothing Span 20, 79",
  ylim = c(0, pi)
)
abline(h = 4 / pi, col = 'blue')
```



From these choices, it seems like the "best" was the one without the logarithm at span `99`. This makes sense, because at $N-1$ the smoothing window is as close as possible to just the sample mean. Because this is a white noise process and the true spectrum is constant, this approch (while slightly silly) looks best. It would not perform well to approximate any process whose spectrum is changing over time; however, the unique property of the WN process of having a constant spectral density means that the closer our smoother gets to simply being the sample mean, the better.

Part D

Suppose we know that the relationship between the periodogram $I(\omega)$ and the spectrum $f(\omega)$ is

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

Taking the log of this relationship, we would observe no change in the error:

$$\frac{2I(\omega)}{f(\omega)} \overset{\text{log-transform}}{\longrightarrow} \log(2) + \log(I(\omega)) - \log(f(\omega)) + \mathrm{V}$$

For some constant V representing the log transform of the random variable.

Now, under this transformation, taking logs has no difference on the error between the two, because the logarithm is a monotonic function. In other words:

$$\forall a \in \mathbb{R}, \forall b \in \mathbb{R},\ a < b \implies \log(a) < \log(b)$$

So, directly from the above one can conclude that the error $I(\omega) - f(\omega)$ will be relatively unaffected by log transformation due to monotonicity, save for a potential change in magnitude.

This is supported by the previous section, where the relative distance of the transformed $I(\omega)$ values to the true spectrum $f(\omega)$ will not experience a "sign change" - reflected in the close similarity of the logarithmic versions (blue lines) and non-log versions (red lines) in shape and scale. **Crucially**, however, prior to a logarithmic transform the error would be multiplicative, and after the log-transform the error is additive. This means that we would likely see differences in magnitude pre and post log transform, while the information conveyed remains relatively the same. Logarithms are extremely useful in situations like these to avoid overflow or underflow in computer-based calculations due to this multiplicative vs. additive factor.

Question 2

Let $\{X_t\}_{t\in\mathbb{Z}}$ be defined by:

$$X_t = Z_t - 0.9Z_{t-1}$$

For $\{X_t\}_{t\in\mathbb{Z}}$, the spectral density is:

$$f(\omega) = \frac{7.24}{\pi}\left(1 - \frac{1.8\cos(\omega)}{1.81}\right), \text{ for } \omega \in (0, 1)$$

We define this function below in `R`:

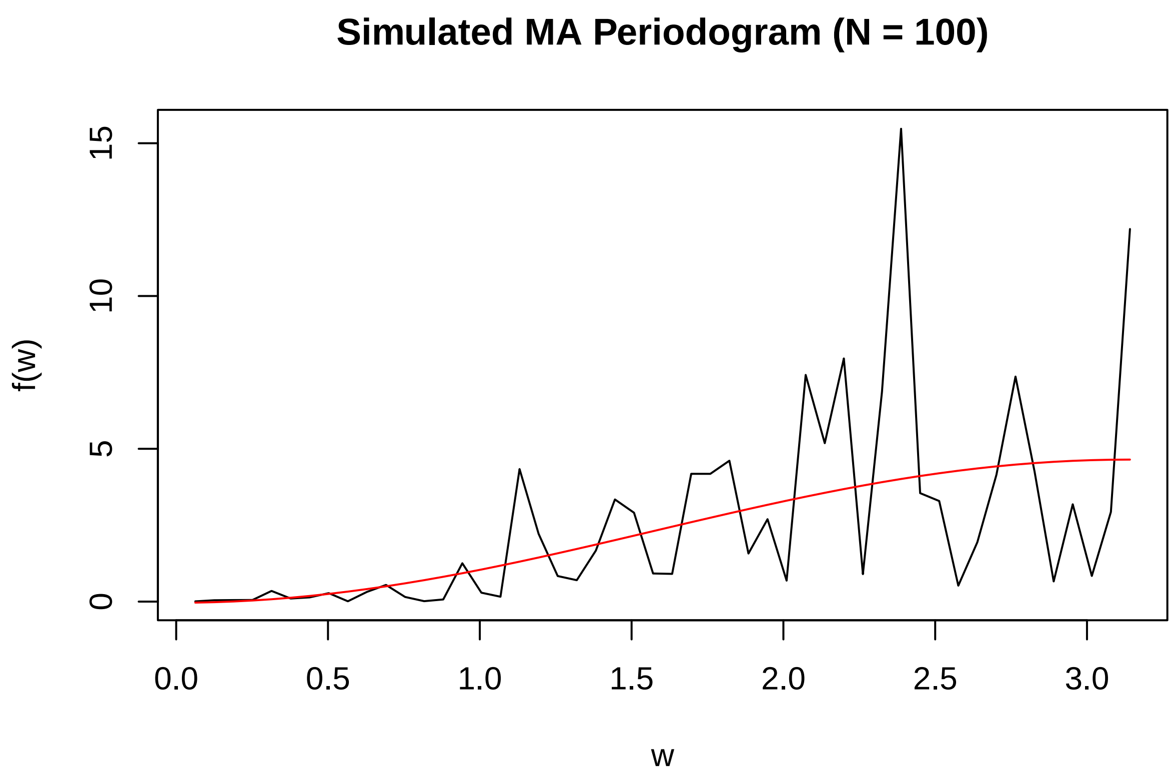
```
f = function(w){
  (7.24)/pi * (1 - (1.84*cos(w))/1.81)
}
```

Part A

Use the `arima.sim` command, or otherwise, to simulate a series of length 100 from $\{X_t\}_{t\in\mathbb{Z}}$. Use `spec.pgram` to create and plot the periodogram for your sample. Then smooth for spans 5, 15 and (7, 5).

Raw Periodogram

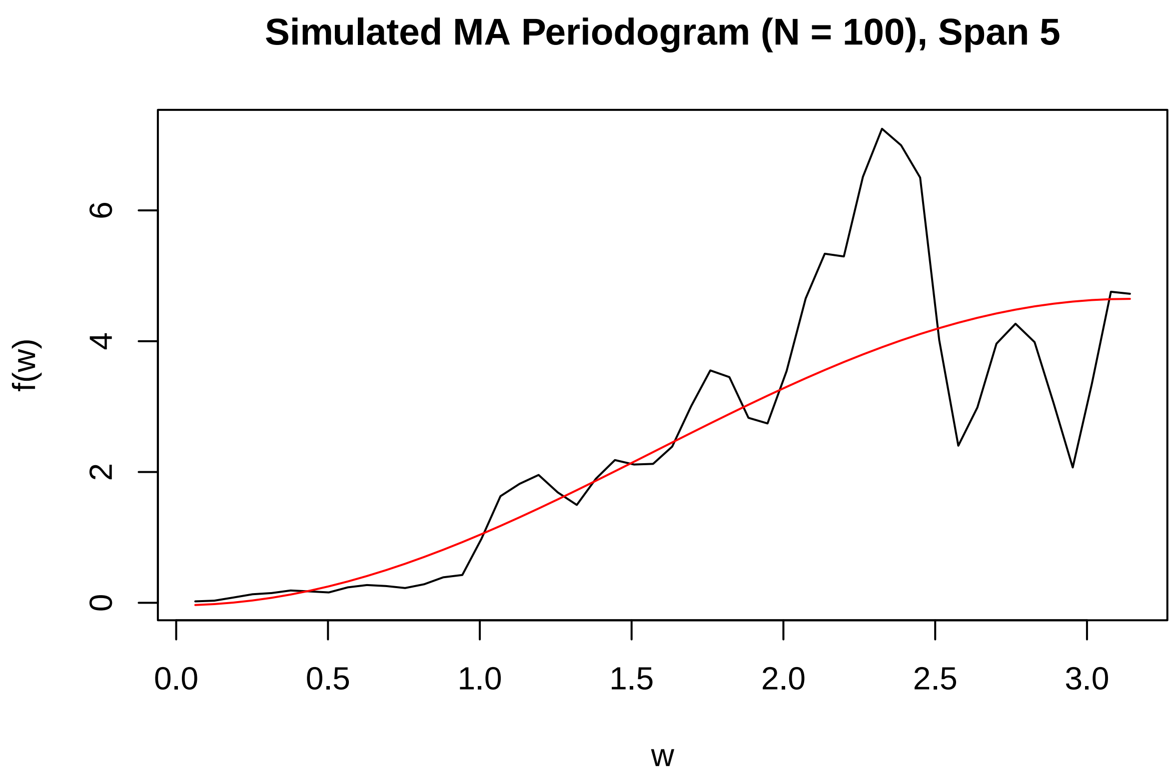
```
set.seed(1924)
MA = arima.sim(model = list(ma = -0.9),
               n = 100,
               sd = 2)
f_w_MA = spec.pgram(MA, log = "no", plot = F)
plot(
  y = f_w_MA$spec / pi,
  x = f_w_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100)"
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



Varying Smoothing Spans

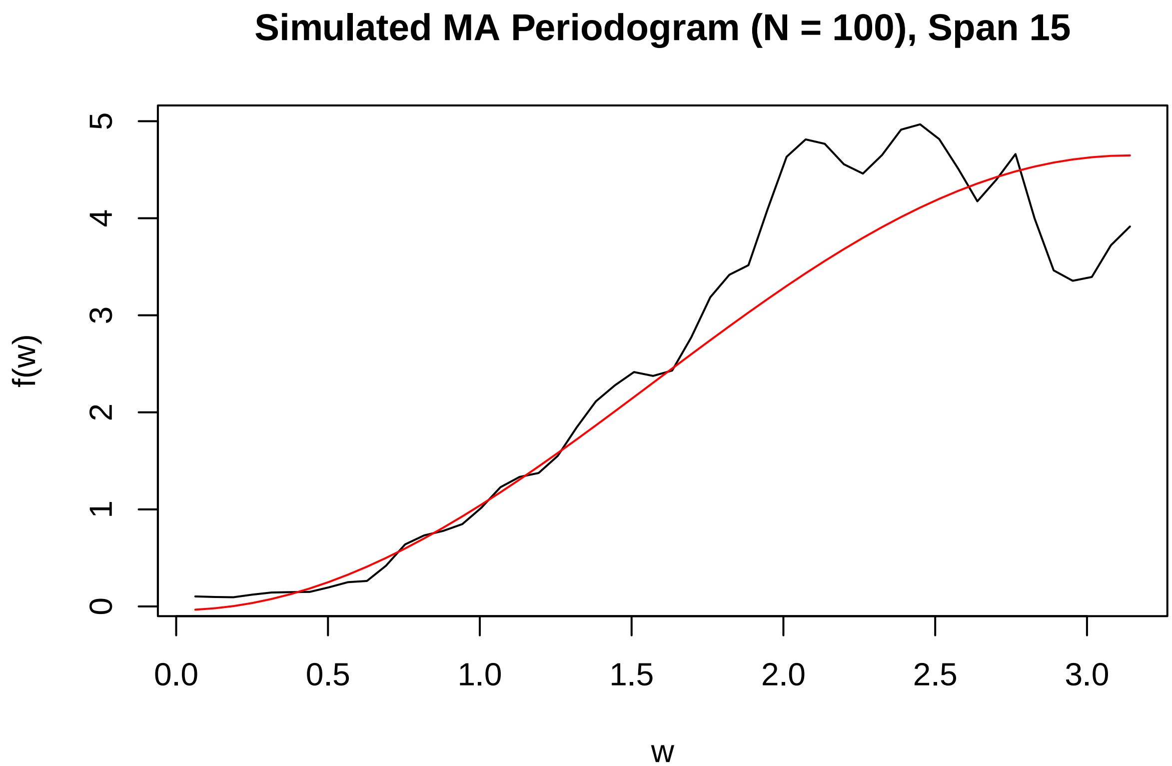
Span 5 Periodogram

```
set.seed(1924)
f_i_MA = spec.pgram(MA, log = "no", plot = F, spans = c(5))
plot(
  y = f_i_MA$spec / pi,
  x = f_i_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100), Span 5"
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



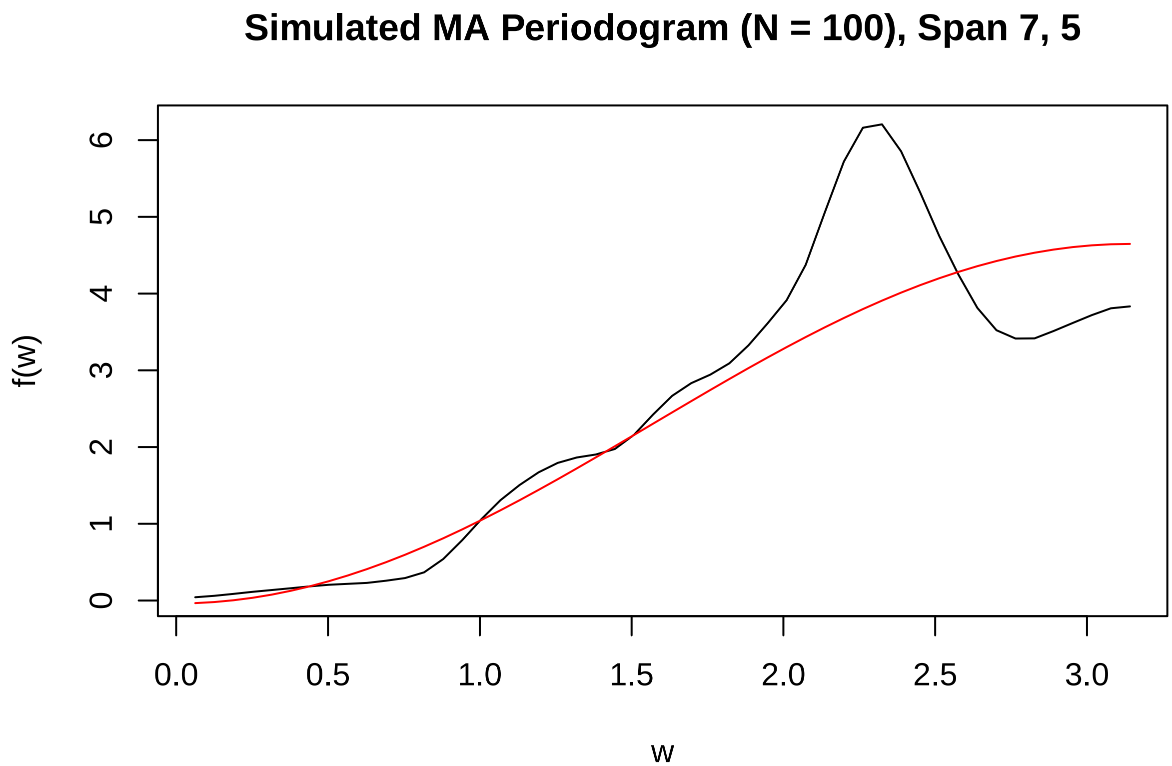
Span 15 Periodogram

```
set.seed(1924)
f_i_MA = spec.pgram(MA, log = "no", plot = F, spans = c(15))
plot(
  y = f_i_MA$spec / pi,
  x = f_i_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100), Span 15"
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



Span 7, 5 Periodogram

```
set.seed(1924)
f_i_MA = spec.pgram(MA, log = "no", plot = F, spans = c(7, 5))
plot(
  y = f_i_MA$spec / pi,
  x = f_i_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100), Span 7, 5"
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



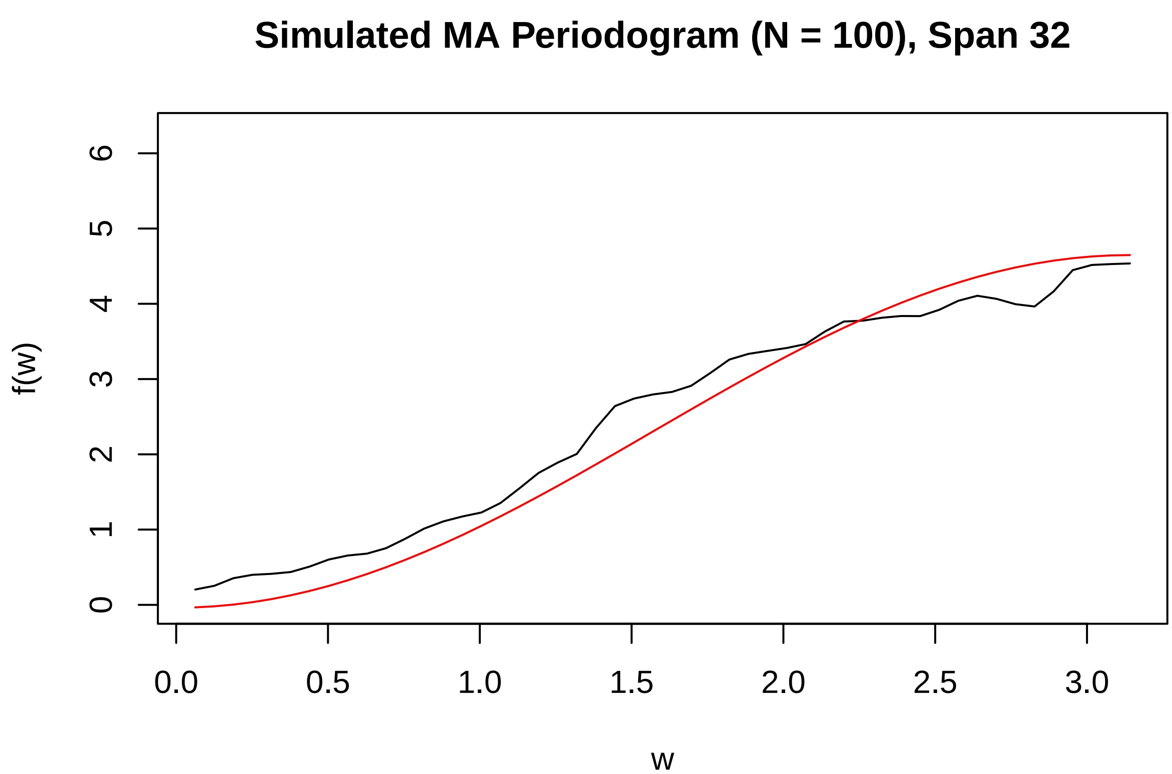
Comments: We note that in the original raw periodogram, the observed $I(\omega)$ values appear to roughly surround the true spectral density; however, there is a lot of noise. As we increase the one-dimensional smoothing span (as seen in plots **i** and **ii**), we see the smoothed periodogram begins to more closely surround the true spectrum, with decreased noise. Interestingly, the two-dimensional `c(7,5)` span shows a "peak" in the spectrum not as noticeable in the `15` span, suggesting that it may not be as effective of a fit in this case. However, the two-dimensional smoother still appears to be a smoother curve overall. The `5` span appears to be under-smoothed, while in this case I would argue the `c(7, 5)` span is more over-smoothed relative to the WN process in the previous question. While the increased dimensionality in the spans seems to impact the smoothing more than a magnitude shift in the one-dimensional span, this doesn't necessarily imply a better overall fit, as we see here. The variance of the `7, 5` span is more obviously lower than the others, but so, too, is its bias due to the well-known bias/variance tradeoff. As always, the "true" best span to use is context-dependent.

Part C

In the one-dimensional case, we iterated through multiple options and found that a span of `32` approximated well.

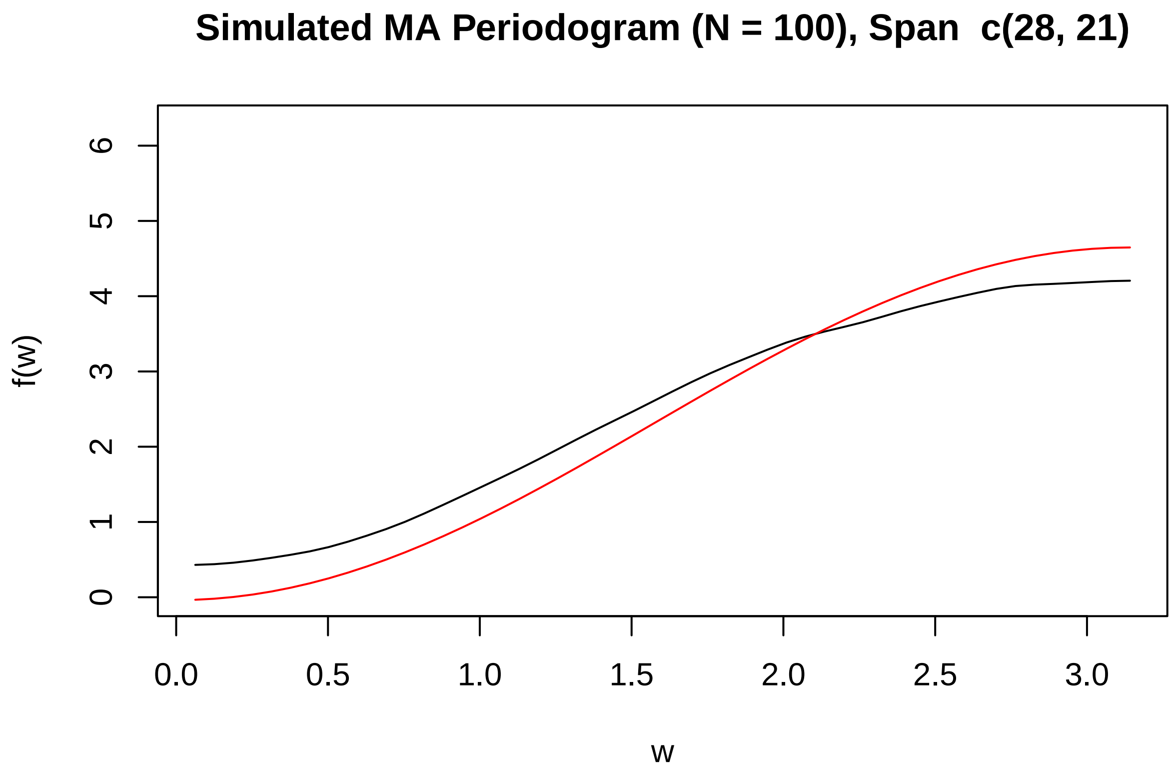
```
set.seed(1924)
f_i_MA = spec.pgram(MA, log = "no", plot = F, spans = c(32))
plot(
  y = f_i_MA$spec / pi,
  x = f_i_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100), Span 32",
```

```
  ylim = c(0, 2*pi)
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



In the double-smoothing case, we found that `c(21, 33)` approximated well.

```
set.seed(1924)
f_i_MA = spec.pgram(MA, log = "no", plot = F, spans = c(21, 33))
plot(
  y = f_i_MA$spec / pi,
  x = f_i_MA$freq * (2 * pi),
  type = 'l',
  ylab = "f(w)",
  xlab = "w",
  main = "Simulated MA Periodogram (N = 100), Span  c(28, 21)",
  ylim = c(0, 2*pi)
)
lines(
  x = f_w_MA$freq * (2 * pi),
  y = f(f_w_MA$freq * (2 * pi)),
  col = 'red'
)
```



Comments: We found that generally, larger values of span approximated the true $f(\omega)$ well. In particular, the double-smoothing case seemed to do an excellent job of capturing the true shape of the spectral density compared to the noisier single-smoothing. This example highlights some of the potential advantages of using double-smoothing to capture the shape of $f(\omega)$, since we know the true density is a smooth curve - this smoothness may not be captured by the single span adjustment.