STAT 443: Lab 11

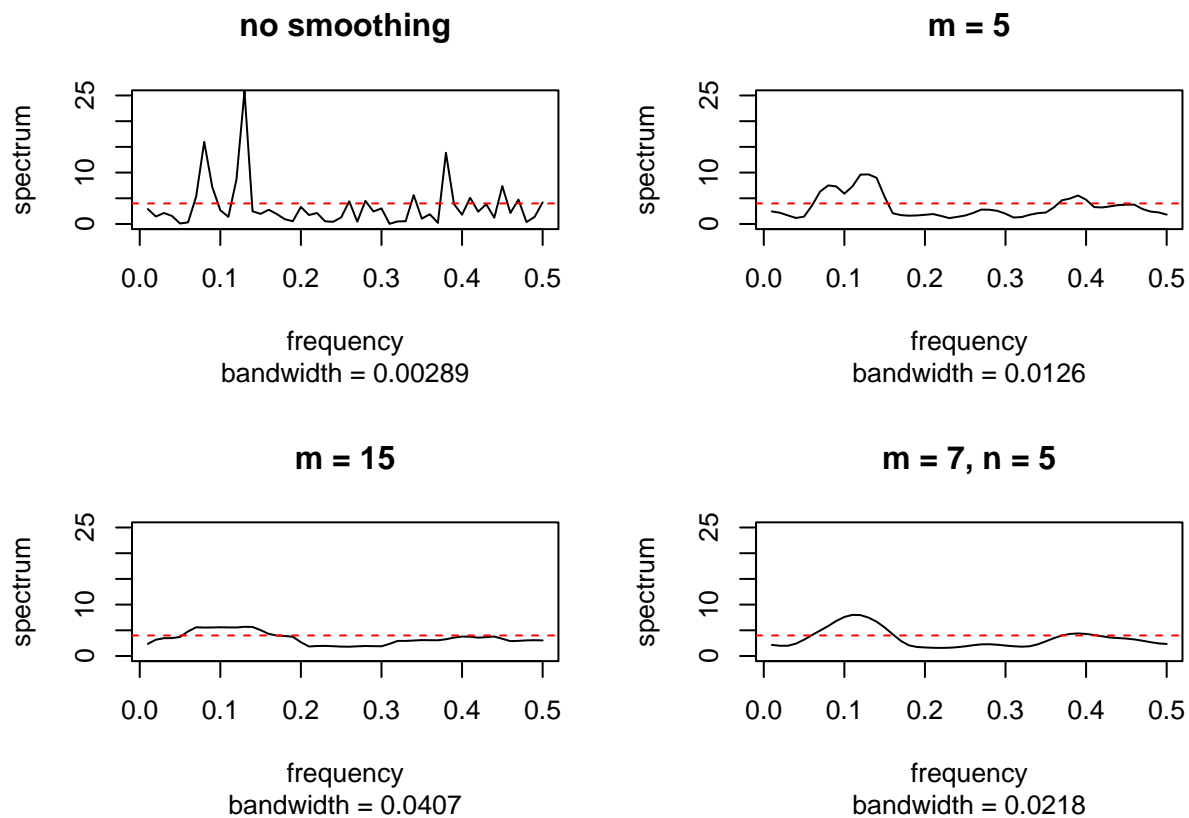## Wenxuan Zan (61336194)

5 April, 2023
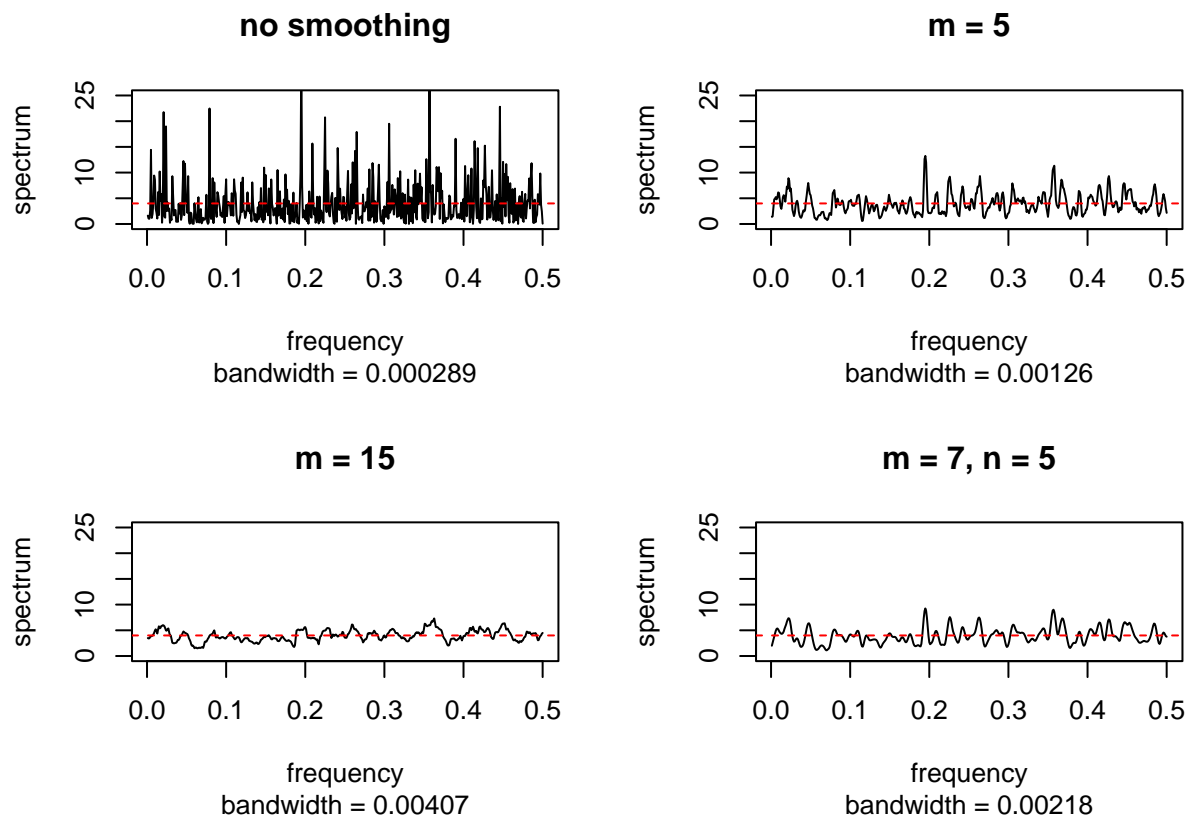
Question 1

a)

```
set.seed(443)
par(mfrow = c(2,2))
wn04 <- arima.sim(n = 100,
                  model = list(),
                  sd = sqrt(4))
spec.pgram(wn04,log = "no",ylim = c(0,25),main = "no smoothing")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(5),ylim = c(0,25), log = "no",main = "m = 5")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(15),ylim = c(0,25), log = "no",main = "m = 15")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(7,5),ylim = c(0,25), log = "no",main = "m = 7, n = 5")
abline(h = 4,col = "red", lty = "dashed")
```

no smoothing

frequency
bandwidth = 0.00289

m = 5

frequency
bandwidth = 0.0126

m = 15

frequency
bandwidth = 0.0407

m = 7, n = 5

frequency
bandwidth = 0.0218

The raw periodogram is very noisy such that there are many spikes around the true variance, and smoothing the original data improves the periodogram in terms of noise reduction. It appears that using a smoothing filter $m = 5$ still result in a noise periodogram. And the using a double smoothing filter with $m = 7, n = 5$ result in a periodogram that deviates too much from the true spectral density function (the red dotted line). Using the $m = 15$ smoothing fitler resulted in the best periodogram that closely resembles the true spectrum.

b)

```
par(mfrow = c(2,2))
wn04 <- arima.sim(n = 1000,
                  model = list(),
                  sd = sqrt(4))
spec.pgram(wn04,log = "no",ylim = c(0,25),main = "no smoothing")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(5),ylim = c(0,25), log = "no",main = "m = 5")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(15),ylim = c(0,25), log = "no",main = "m = 15")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(7,5),ylim = c(0,25), log = "no",main = "m = 7, n = 5")
abline(h = 4,col = "red", lty = "dashed")
```

no smoothing

frequency
bandwidth = 0.000289

m = 5

frequency
bandwidth = 0.00126

m = 15

frequency
bandwidth = 0.00407

m = 7, n = 5

frequency
bandwidth = 0.00218

- When increasing the sample size from 100 to 1000, there is no improvement in the periodograms. In fact, all periodograms have become noiser compare to their counterparts in part a).

- Using $m = 15$ smoothing filter resulted in a periodogram that closely resembles the true spectrum (the red dashed line) compared with using smoothing filter $m = 5$ and $m = 7, n = 5$.
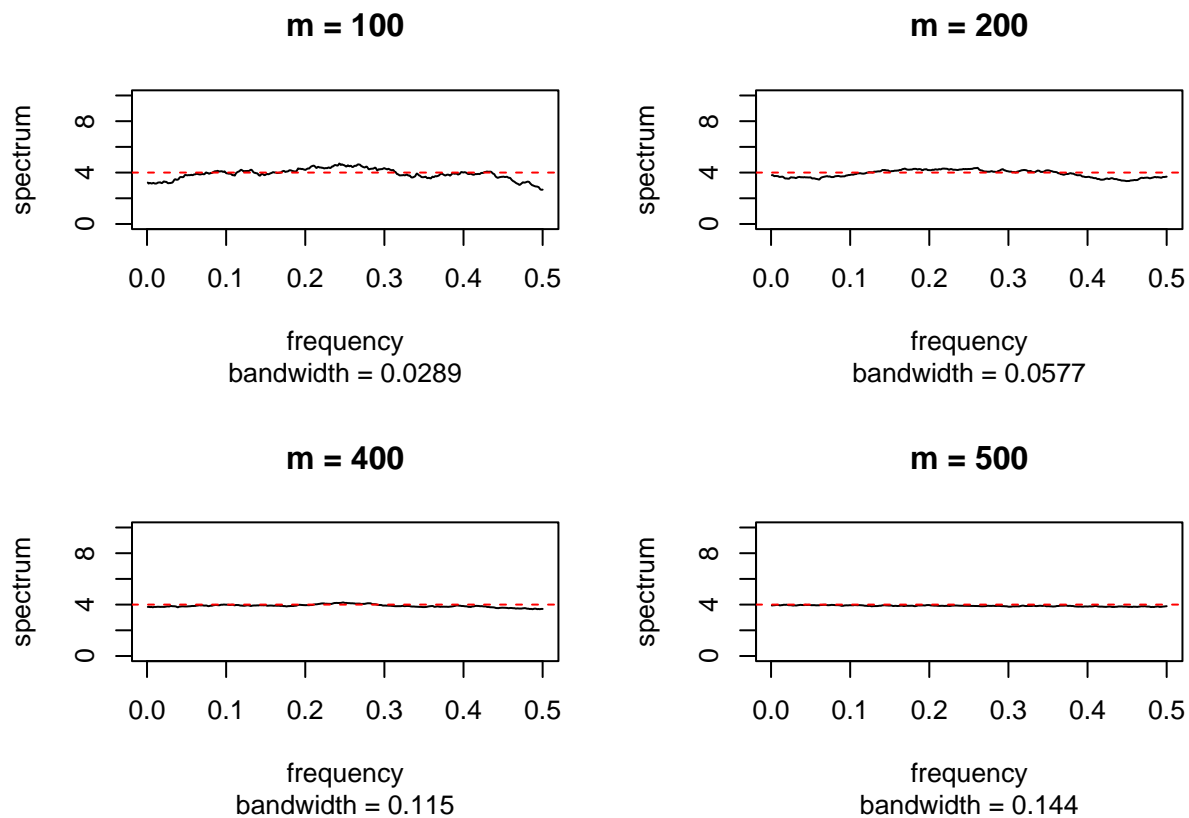
c)

```
set.seed(433)
par(mfrow = c(2,2))
wn04 <- arima.sim(n = 1000,
                  model = list(),
                  sd = sqrt(4))
spec.pgram(wn04,spans = c(50), log = "no",main = "m = 50")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(50),log = "yes", main = "m = 50 with log transformation")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(500), log = "no",main = "m = 500")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(500), log = "yes", main = "m = 500 with log transformation")
abline(h = 4,col = "red", lty = "dashed")
```

3

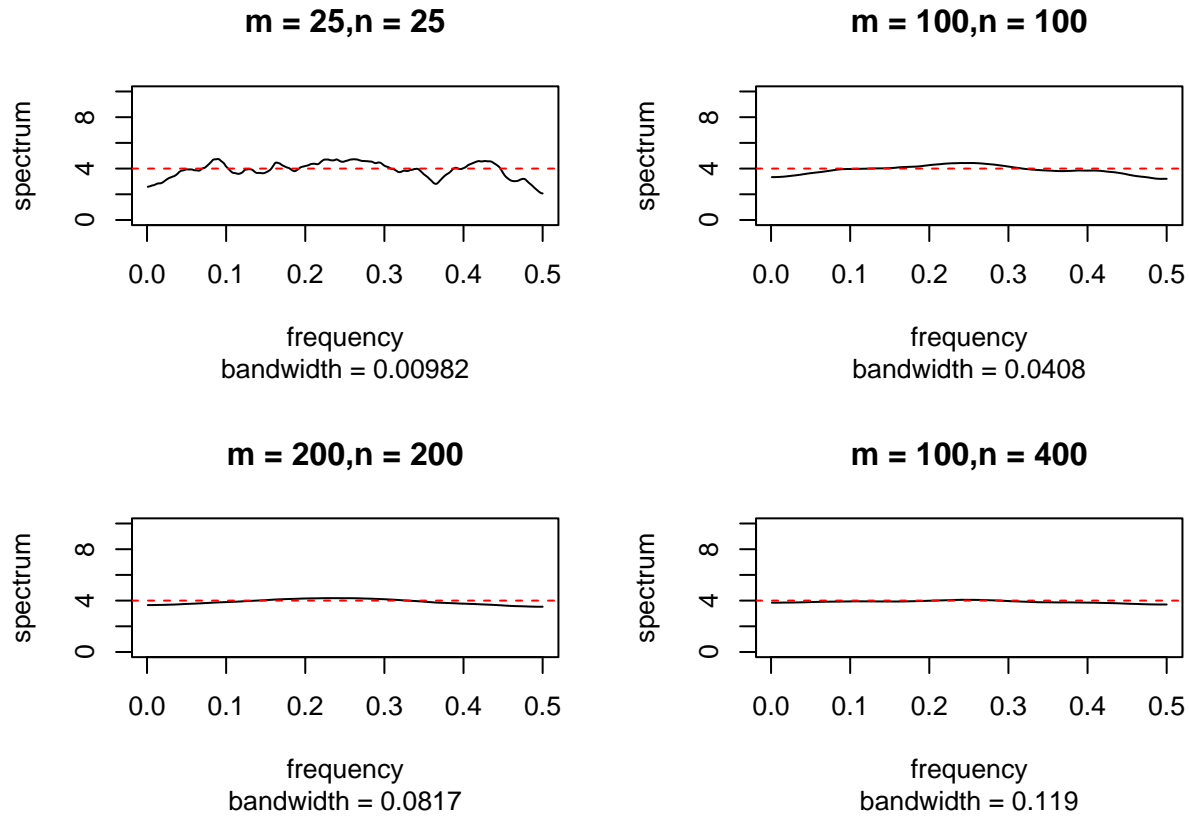|			
m = 50		**m = 50 with log transformation**	
m = 500		**m = 500 with log transformation**	
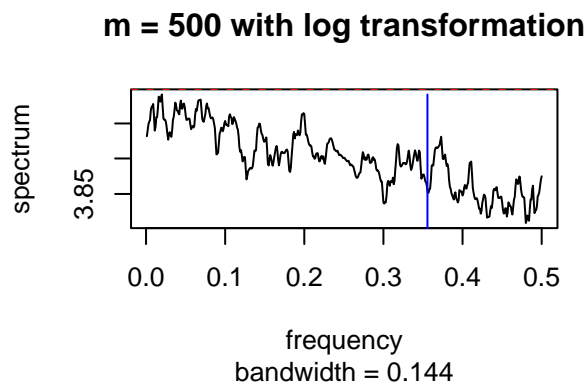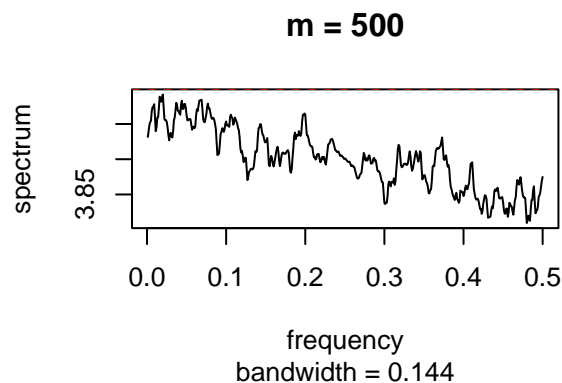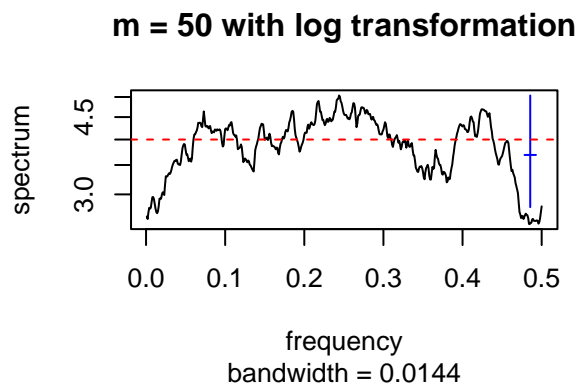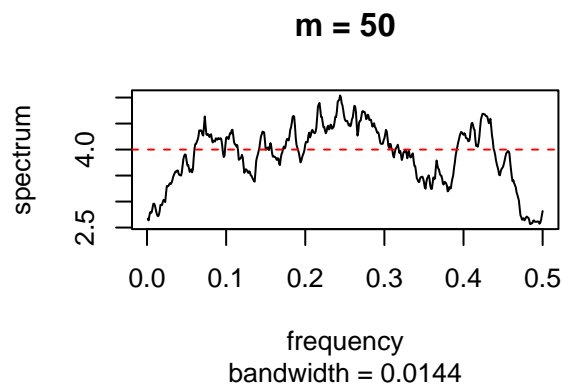
- It appears that taking a log transformation did not result in any observable changes to the periodogram as demonstrated with the above simulations.

```
par(mfrow = c(2,2))
spec.pgram(wn04,spans = c(100),ylim = c(0,10), log = "no",main = "m = 100")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(200),ylim = c(0,10), log = "no",main = "m = 200")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(400),ylim = c(0,10), log = "no",main = "m = 400")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(500),ylim = c(0,10), log = "no",main = "m = 500")
abline(h = 4,col = "red", lty = "dashed")
```

m = 100

m = 200

m = 400

m = 500

frequency
bandwidth = 0.0289

frequency
bandwidth = 0.0577

frequency
bandwidth = 0.115

frequency
bandwidth = 0.144

```
par(mfrow = c(2,2))
spec.pgram(wn04,spans = c(25,25),ylim = c(0,10), log = "no",main = "m = 25,n = 25")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(100,100),ylim = c(0,10), log = "no",main = "m = 100,n = 100")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(200,200),ylim = c(0,10), log = "no",main = "m = 200,n = 200")
abline(h = 4,col = "red", lty = "dashed")
spec.pgram(wn04,spans = c(100,400),ylim = c(0,10), log = "no",main = "m = 100,n = 400")
abline(h = 4,col = "red", lty = "dashed")
```

m = 25,n = 25

frequency
bandwidth = 0.00982

m = 100,n = 100

frequency
bandwidth = 0.0408

m = 200,n = 200

frequency
bandwidth = 0.0817

m = 100,n = 400

frequency
bandwidth = 0.119

- Looking at the above simulations, using $m = 500$ smoothing filter resulted in the best periodogram. The $m = 500$ filter smoothed out majority of the variations and the resulting periodogram closely resmbles the true spectrum.

d)

- Taking a log transformation does not change the error between $I(\omega)$ and $f(\omega)$, so $I(\omega)$ is still unbiased for $f(\omega)$. This is consistent with observations made in c).

Question 2

a)

```r
x <- seq(0,pi,by = 0.01)
y = (7.25/pi)*(1-(1.8*cos(x))/1.81)
set.seed(443)
ma1 <- arima.sim(n = 100,
                 model = list(ma = c(-0.9)),
                 sd = sqrt(4))
par(mfrow = c(2,2))
spec.pgram(ma1,log = "no",ylim = c(0,40),main = "no smoothing")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(5),ylim = c(0,20), log = "no",main = "m = 5")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(15),ylim = c(0,20), log = "no",main = "m = 15")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
```
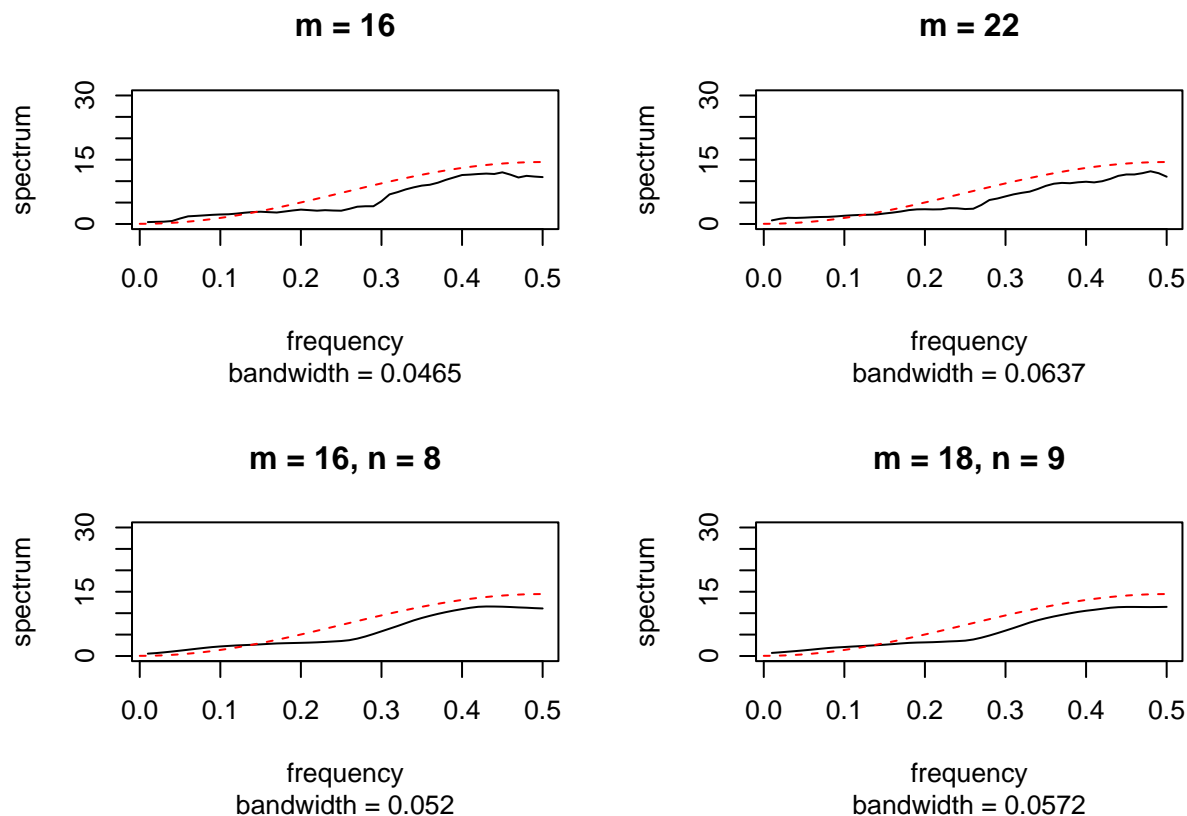
6

```
spec.pgram(ma1,spans = c(7,5),ylim = c(0,20), log = "no",main = "m = 7, n = 5")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
```
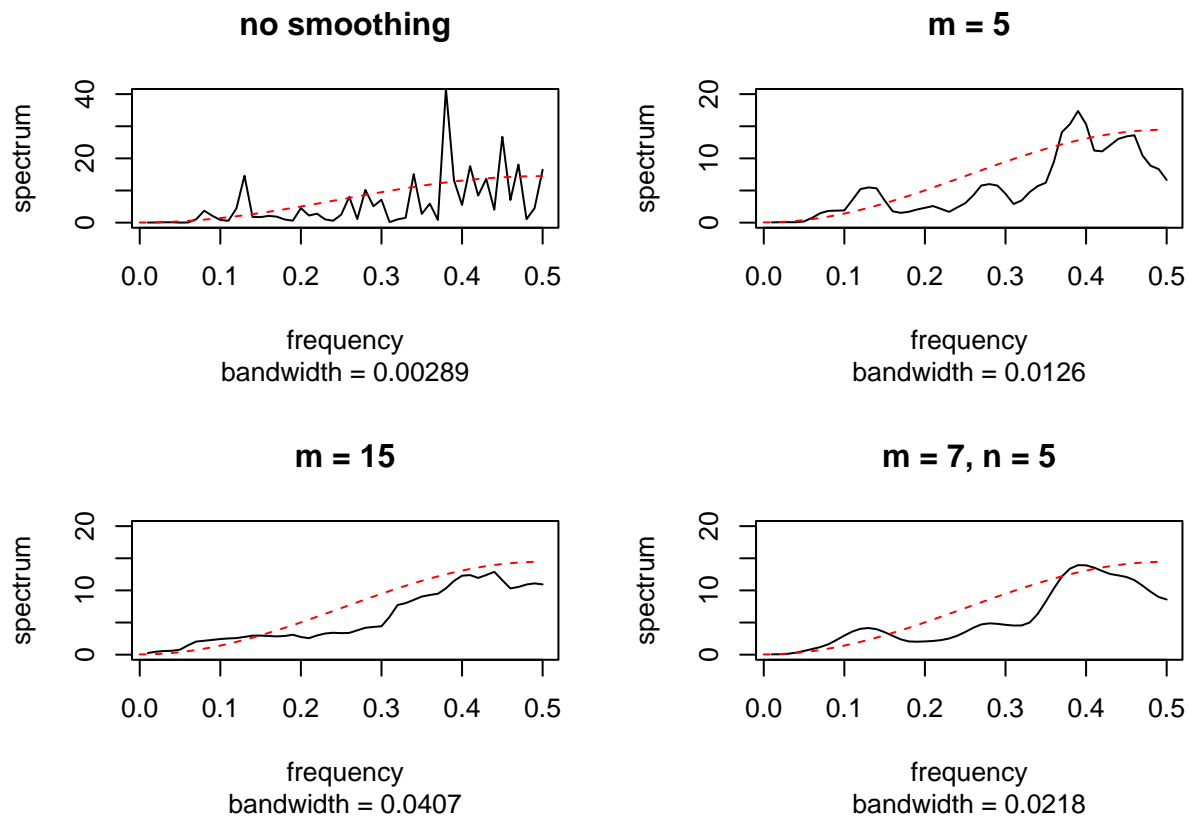

- The original periodogram is very noise and have many peaks. Whereas the smoothed series retains the overall shape of the original series and are more similar in shape to the true spectrum of a $MA(1)$ process (red dashed line). The smoothed series with $m = 15$ performed the best out of the three smoothed series since its shape resembles more of the true spectrum.
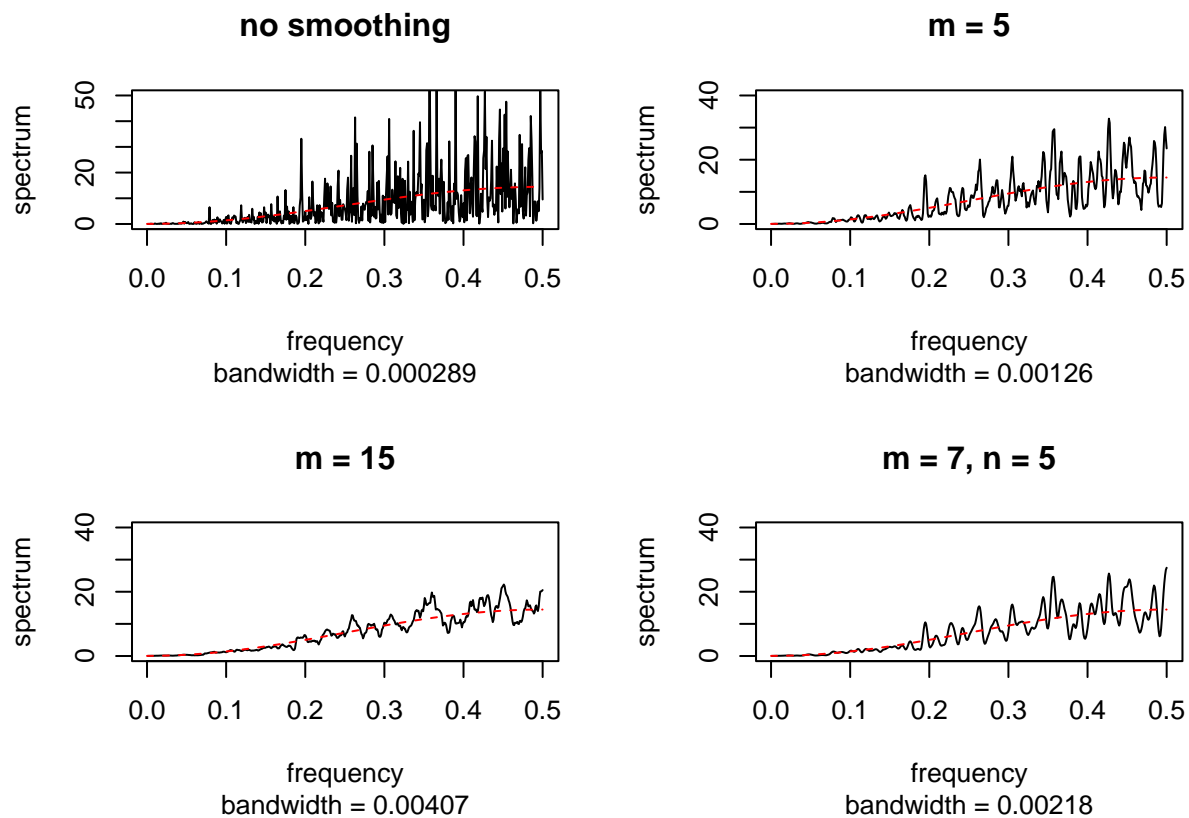
b)

```
par(mfrow = c(2,2))
spec.pgram(ma1,spans = c(16),ylim = c(0,30), log = "no",main = "m = 16")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(22),ylim = c(0,30), log = "no",main = "m = 22")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(16,8),ylim = c(0,30), log = "no",main = "m = 16, n = 8")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(18,9),ylim = c(0,30), log = "no",main = "m = 18, n = 9")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
```

7

m = 16
frequency
bandwidth = 0.0465

m = 22
frequency
bandwidth = 0.0637

m = 16, n = 8
frequency
bandwidth = 0.052

m = 18, n = 9
frequency
bandwidth = 0.0572

- It appears that when $m = 22$ the single smoother give the best periodogram, as we can see the periodogram is closer in shape to the true spectrum; when $m = 18, n = 9$ the double smoother give the best periodogram as it resembles the true spectrum more.
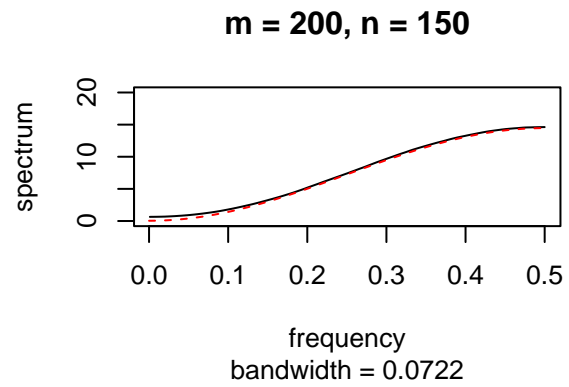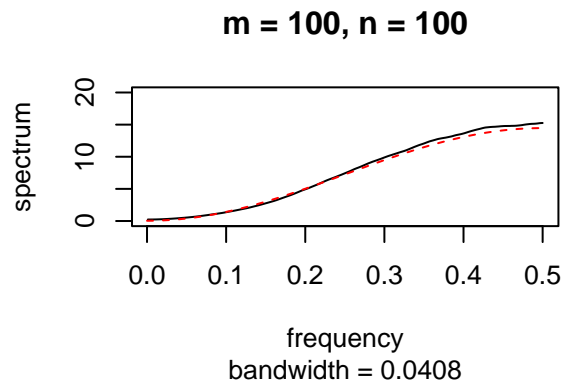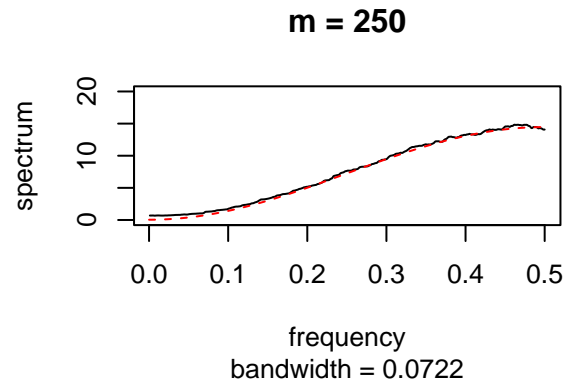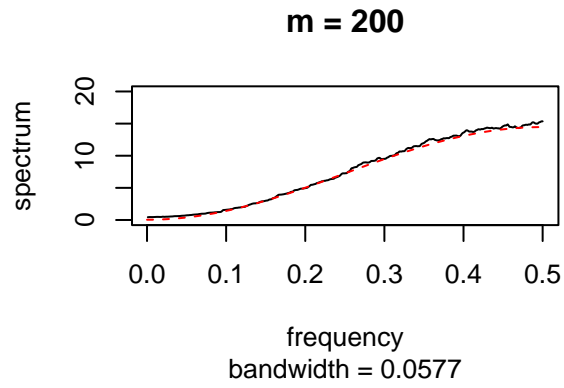
c)

```r
x <- seq(0,pi,by = 0.001)
y = (7.25/pi)*(1-(1.8*cos(x))/1.81)
set.seed(443)
ma1 <- arima.sim(n = 1000,
                 model = list(ma = c(-0.9)),
                 sd = sqrt(4))
par(mfrow = c(2,2))
spec.pgram(ma1,log = "no",ylim = c(0,50),main = "no smoothing")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(5),ylim = c(0,40), log = "no",main = "m = 5")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(15),ylim = c(0,40), log = "no",main = "m = 15")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(7,5),ylim = c(0,40), log = "no",main = "m = 7, n = 5")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
```

no smoothing / **m = 5** / **m = 15** / **m = 7, n = 5**

- As sample size increase the periodogram becomes more noise as before, and we also observe that as we apply a smoothing filter with longer span, the variance is greatly reduced. If we compare the smoothed periodogram to the original periodogram, we can see that $m = 15$ performed the best as it has the least noise and its shape is similar to the true spectrum.

d)

```
m1 = 350
m2 = 200
n = 150
x <- seq(0,pi,by = 0.001)
y = (7.25/pi)*(1-(1.8*cos(x))/1.81)
par(mfrow = c(2,2))
spec.pgram(ma1,spans = c(200),ylim = c(0,20), log = "no",main = "m = 200")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(250),ylim = c(0,20), log = "no",main = "m = 250")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(100,100),ylim = c(0,20), log = "no",main = "m = 100, n = 100")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
spec.pgram(ma1,spans = c(200,150),ylim = c(0,20), log = "no",main = "m = 200, n = 150")
lines(x/(2*pi),y*pi,col = "red",lty="dashed")
```

m = 200

spectrum

frequency
bandwidth = 0.0577

m = 250

spectrum

frequency
bandwidth = 0.0722

m = 100, n = 100

spectrum

frequency
bandwidth = 0.0408

m = 200, n = 150

spectrum

frequency
bandwidth = 0.0722

- When $n = 1000$, it appears that when $m = 250$ the single smoother give the best periodogram, and when $m = 200, n = 150$ the double smoother give the best periodogram. In both cases, the smoothed periodogram follows the true spectrum very closely. It appears that as the sample size increases, we need to increase the size of m, n in our smoothing process as well.