

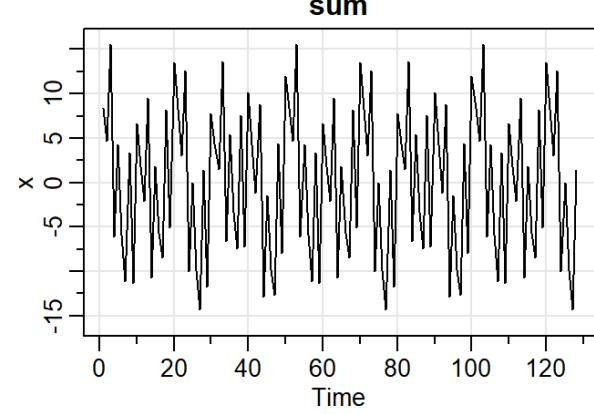
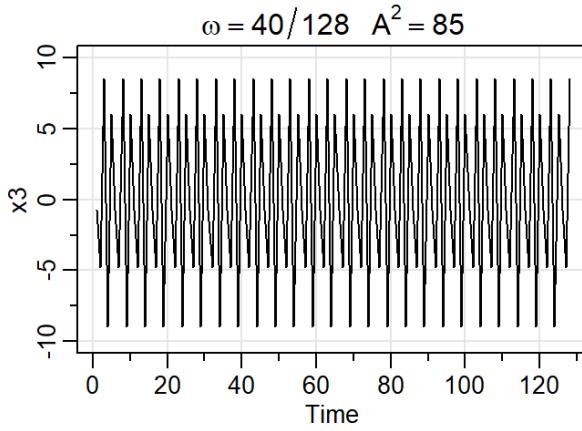
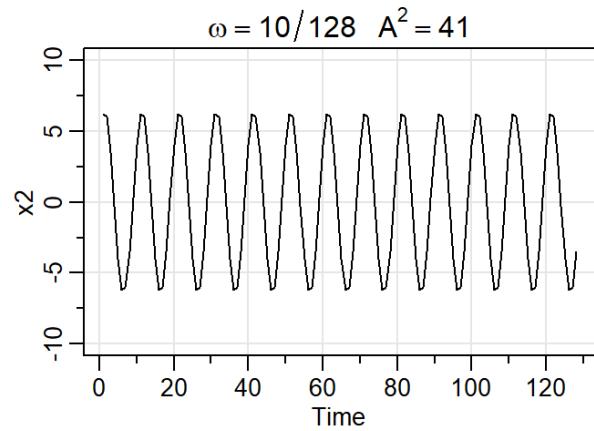
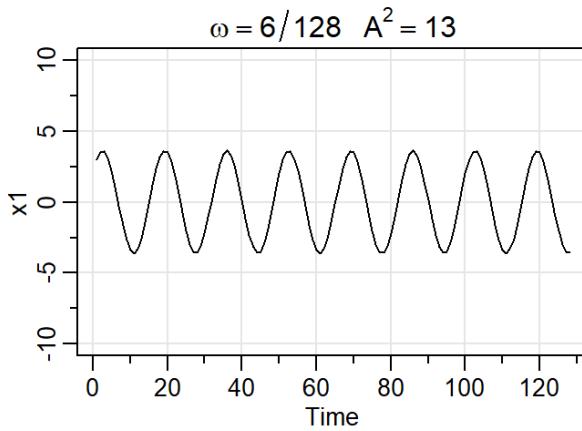
Chapter 6/7 Homework

Orly Olbum

6.1

Repeat the simulations and analyses in Example 6.1 and Example 6.2 with the following changes: (a) Change the sample size to $n = 128$ and generate and plot the same series as in Example 6.1: $xt1 = 2\cos(2\pi \cdot 0.06t) + 3\sin(2\pi \cdot 0.06t)$, $xt2 = 4\cos(2\pi \cdot 1.0t) + 5\sin(2\pi \cdot 1.0t)$, $xt3 = 6\cos(2\pi \cdot 40t) + 7\sin(2\pi \cdot 40t)$, $xt = xt1 + xt2 + xt3$. What is the major difference between these series and the series generated in Example 6.1? (Hint: The answer is fundamental. But if your answer is the series are longer, you may be punished severely.)

```
x1 = 2*cos(2*pi*1:128*6/100) + 3*sin(2*pi*1:128*6/100)
x2 = 4*cos(2*pi*1:128*10/100) + 5*sin(2*pi*1:128*10/100)
x3 = 6*cos(2*pi*1:128*40/100) + 7*sin(2*pi*1:128*40/100)
x = x1 + x2 + x3
par(mfrow = c(2, 2))
tsplot(x1, ylim = c(-10, 10), main = expression(omega == 6/128 ~~~ A^2 == 13))
tsplot(x2, ylim = c(-10, 10), main = expression(omega == 10/128 ~~~ A^2 == 41))
tsplot(x3, ylim = c(-10, 10), main = expression(omega == 40/128 ~~~ A^2 == 85))
tsplot(x, ylim = c(-16, 16), main = "sum")
```



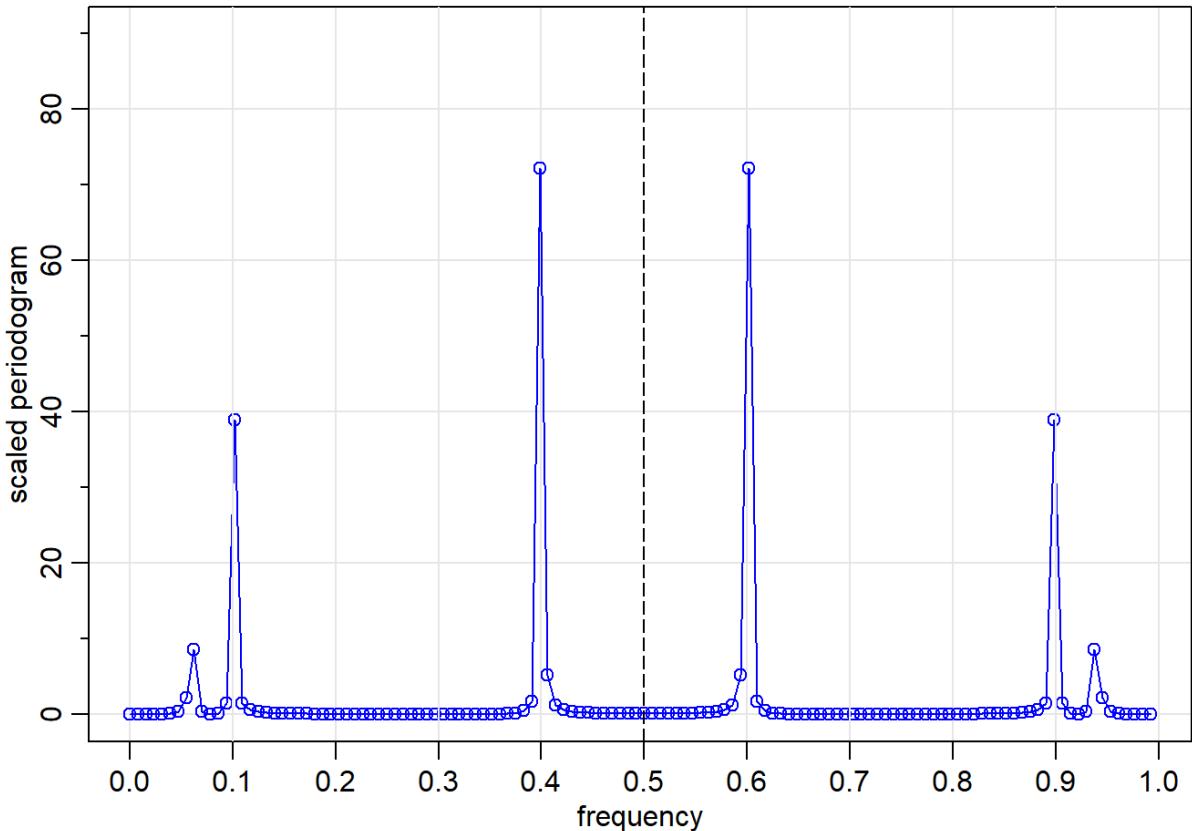
This series is different than the series from Example 6.1 because the fundamental frequencies are larger than those of the $n=100$ simulation from Example 6.1. Since 128 is even and is not so far away from 100, not much else changes. We will see more in the periodogram.

(b) As in Example 6.2, compute and plot the periodogram of the series, xt , generated in (a) and comment.

```

P = Mod(fft(x)/sqrt(128))^2
sP = (4/128)*P
Fr = 0:127/128
par(mfrow = c(1, 1))
tsplot(Fr, sP, type="o", xlab = "frequency", ylab = "scaled periodogram",
       col = 4, ylim = c(0,90))
abline(v = .5, lty = 5)
abline(v = c(.1, .3, .7, .9), lty = 1, col = gray(.9))
axis(side = 1, at = seq(.1, .9, by = .2))

```



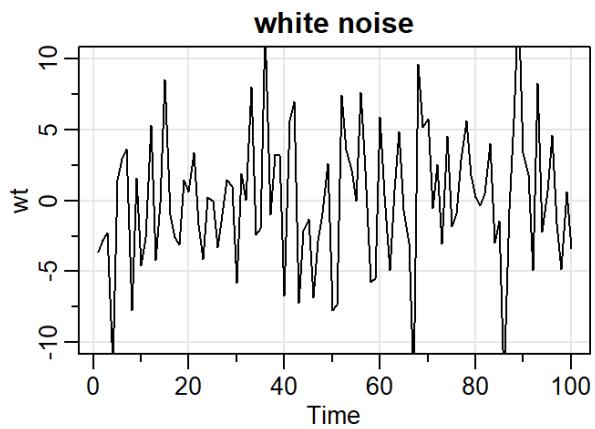
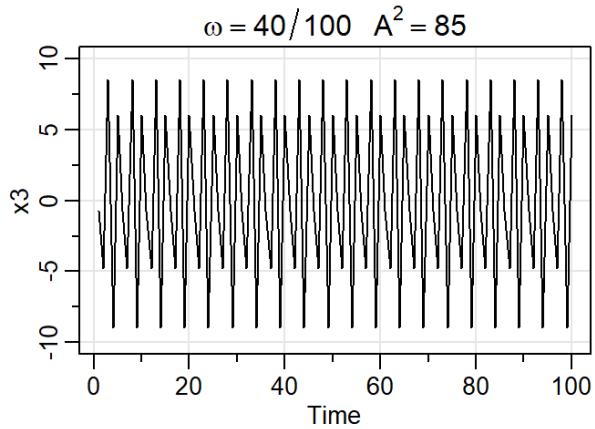
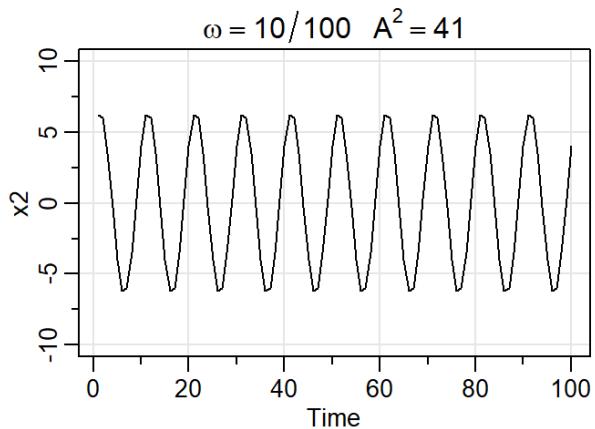
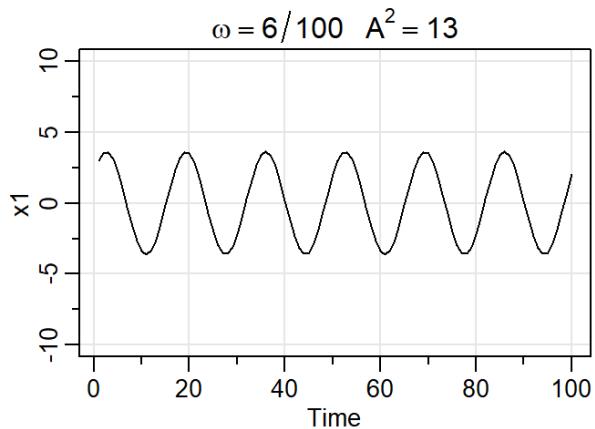
We can clearly see from the periodogram that at the peaks have sharpened (to the extent of the graph not knowing how to show the almost vertical line) and narrowed. This indicates that the frequency has changed to be higher than those of the $n=100$ simulations from Example 6.1.

(c) Repeat the analyses of (a) and (b) but with $n = 100$ (as in Example 6.1), and adding noise to xt ; that is $xt = xt1 + xt2 + xt3 + wt$ where wt are iid $N(0, \text{sigma}_w = 5)$. That is, you should simulate and plot the data, and then plot the periodogram of xt and comment.

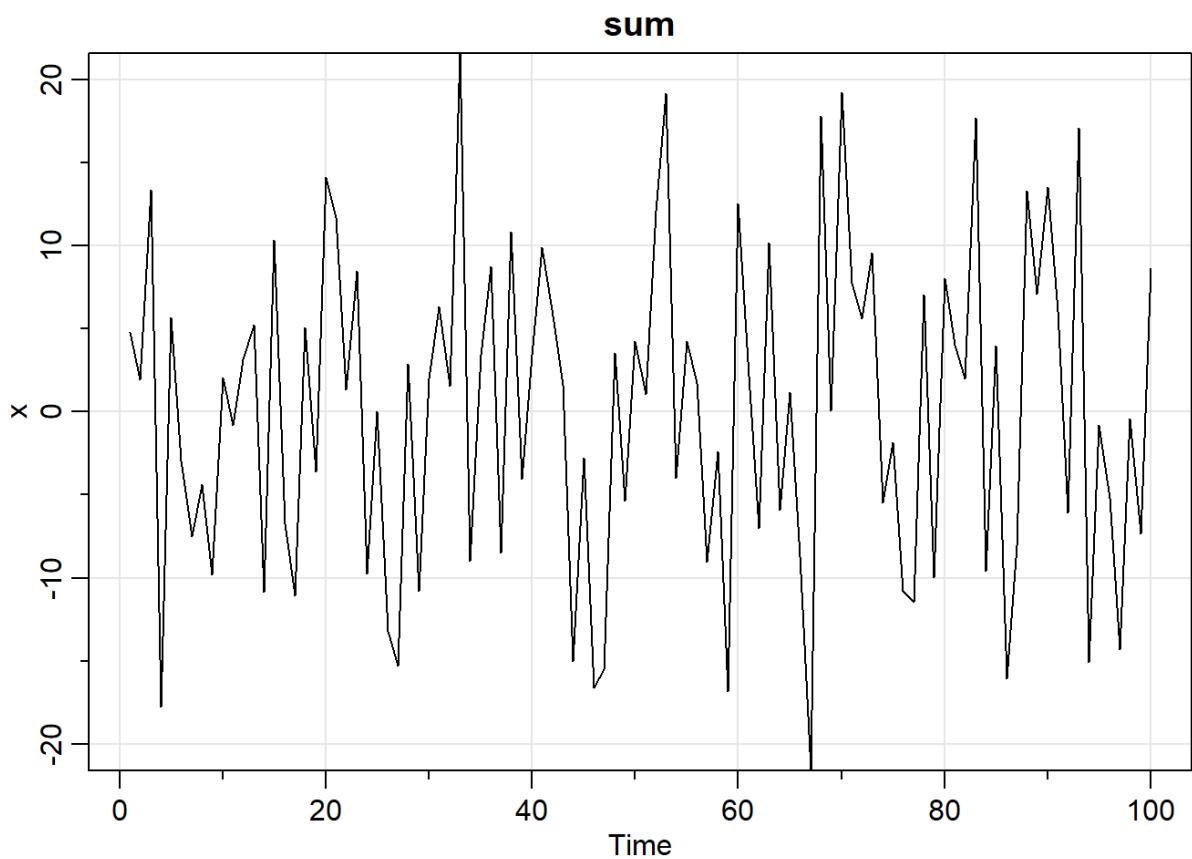
```

x1 = 2*cos(2*pi*1:100*6/100) + 3*sin(2*pi*1:100*6/100)
x2 = 4*cos(2*pi*1:100*10/100) + 5*sin(2*pi*1:100*10/100)
x3 = 6*cos(2*pi*1:100*40/100) + 7*sin(2*pi*1:100*40/100)
wt = rnorm(n = 100, mean = 0, sd = 5)
x = x1 + x2 + x3 + wt
par(mfrow = c(2, 2))
tsplot(x1, ylim = c(-10, 10), main = expression(omega == 6/100 ~~~ A^2 == 13))
tsplot(x2, ylim = c(-10, 10), main = expression(omega == 10/100 ~~~ A^2 == 41))
tsplot(x3, ylim = c(-10, 10), main = expression(omega == 40/100 ~~~ A^2 == 85))
tsplot(wt, ylim = c(-10, 10), main = "white noise")

```



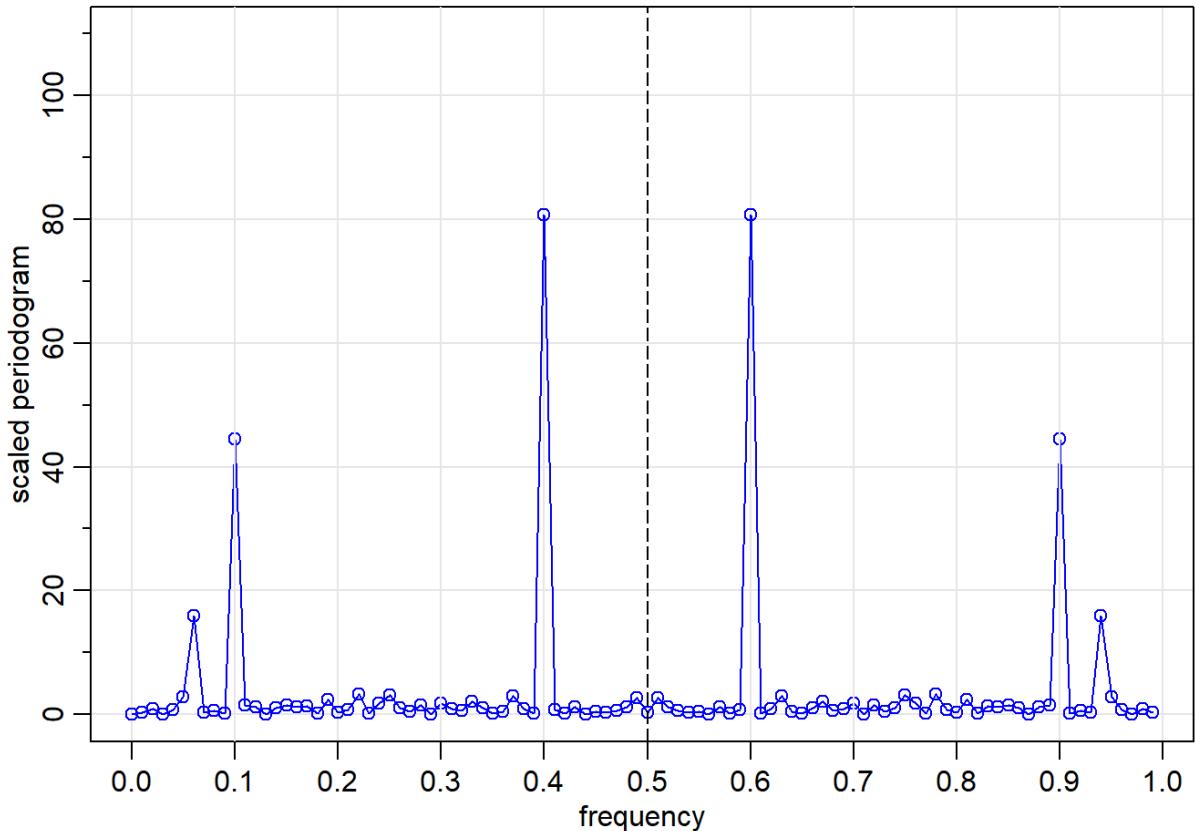
```
par(mfrow = c(1, 1))
tsplot(x, ylim = c(-20, 20), main = "sum")
```



```

P = Mod(fft(x)/sqrt(100))^2 # periodogram
sP = (4/100)*P # scaled periodogram
Fr = 0:99/100 # fundamental frequencies
par(mfrow = c(1, 1))
tsplot(Fr, sP, type = "o", xlab = "frequency", ylab = "scaled periodogram",
       col = 4, ylim = c(0, 110))
abline(v = .5, lty = 5)
abline(v = c(.1, .3, .7, .9), lty = 1, col = gray(.9))
axis(side = 1, at = seq(.1, .9, by = .2))

```



We can see that when we've added white noise, the periodogram is even sharper and narrower, and the periodogram values go much higher at each frequency (note the y-axis limit has changed to accomodate).

6.9

Suppose xt is stationary and we apply two filtering operations in succession, say:

$$y_t = \sum_r a_r x_{t-r} \quad \text{then} \quad z_t = \sum_s b_s y_{t-s}.$$

(a) Use property 6.11 to show the spectrum of the output is:

$$f_z(\omega) = |A(\omega)|^2 |B(\omega)|^2 f_x(\omega),$$

where $A(w)$ and $B(w)$ are the Fourier transforms of the filter sequences a_t and b_t , respectively.

6.9

x_t is stationary, apply filtering

$$y_t = \sum_r a_r x_{t-r} \text{ then } z_t = \sum_s b_s y_{t-s}$$

@ use property 6.11 to show spectrum of output is

$$f_z(w) = |A(w)|^2 |B(w)|^2 f_x(w)$$

where $A(w)$, $B(w)$ are Fourier transforms of a_t & b_t

From property 6.11, we know

$$f_y(w) = |A(w)|^2 f_x(w)$$

and

$$f_z(w) = |B(w)|^2 f_y(w)$$

Plugging in $f_y(w)$ from above,

$$f_z(w) = |B(w)|^2 |A(w)|^2 f_x(w)$$

$$= |A(w)|^2 |B(w)|^2 f_x(w)$$

(b) What would the effect of applying the filter:

$$u_t = x_t - x_{t-12} \text{ followed by } v_t = u_t - u_{t-1}$$

to a time series?

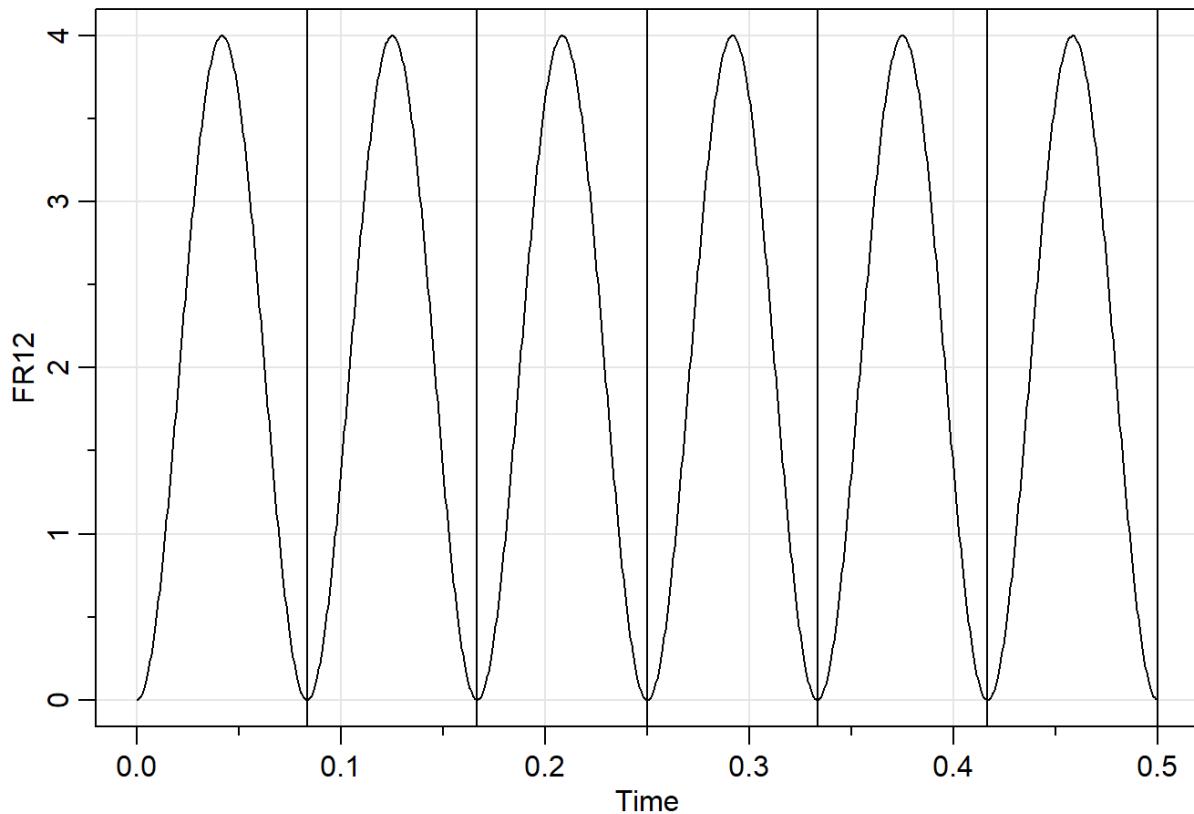
```
w = seq(0, 0.5, length = 1000)
par(mfrow = c(2, 1))
FR12 = abs(1 - exp(2i*12*pi*w))^2
```

By smoothing twice in this manner twice, we are able to see frequencies retained - the first differencing finds frequencies prominent at the 12-cycles, and the second differencing further smooths the data to see the overall frequency. The second difference additionally breaks through higher frequencies to see lower frequencies by removing season trends, and we see that it is retained in the second plot, below.

(c) Plot the frequency responses of the filters associated with u_t and v_t described in (b).

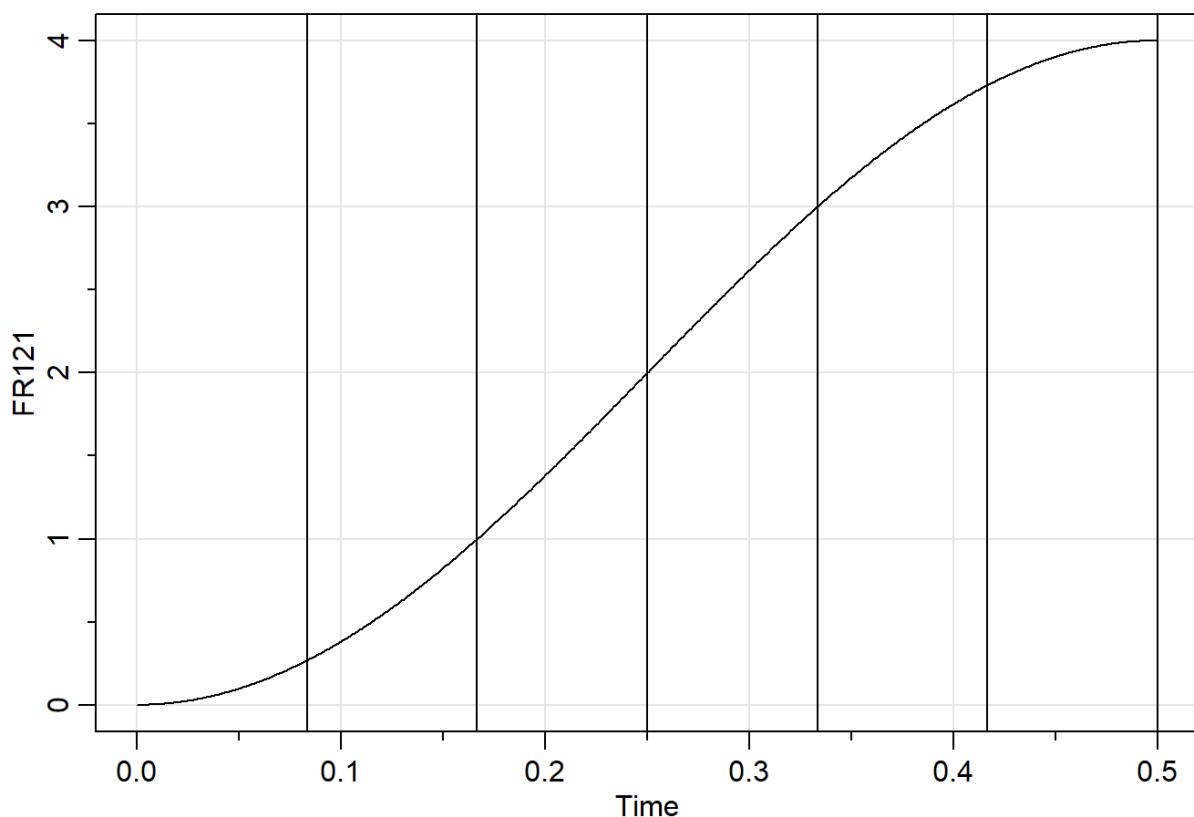
```
tsplot(w, FR12, main = "12th Difference")
abline(v = 1:6/12)
```

12th Difference



```
FR121 = abs(1 - exp(2i*pi*w) - exp(2i*12*pi*w) + exp(2i*12*pi*w))^2  
tsplot(w, FR121, main = "1st Diff and 12th Diff")  
abline(v = 1:6/12)
```

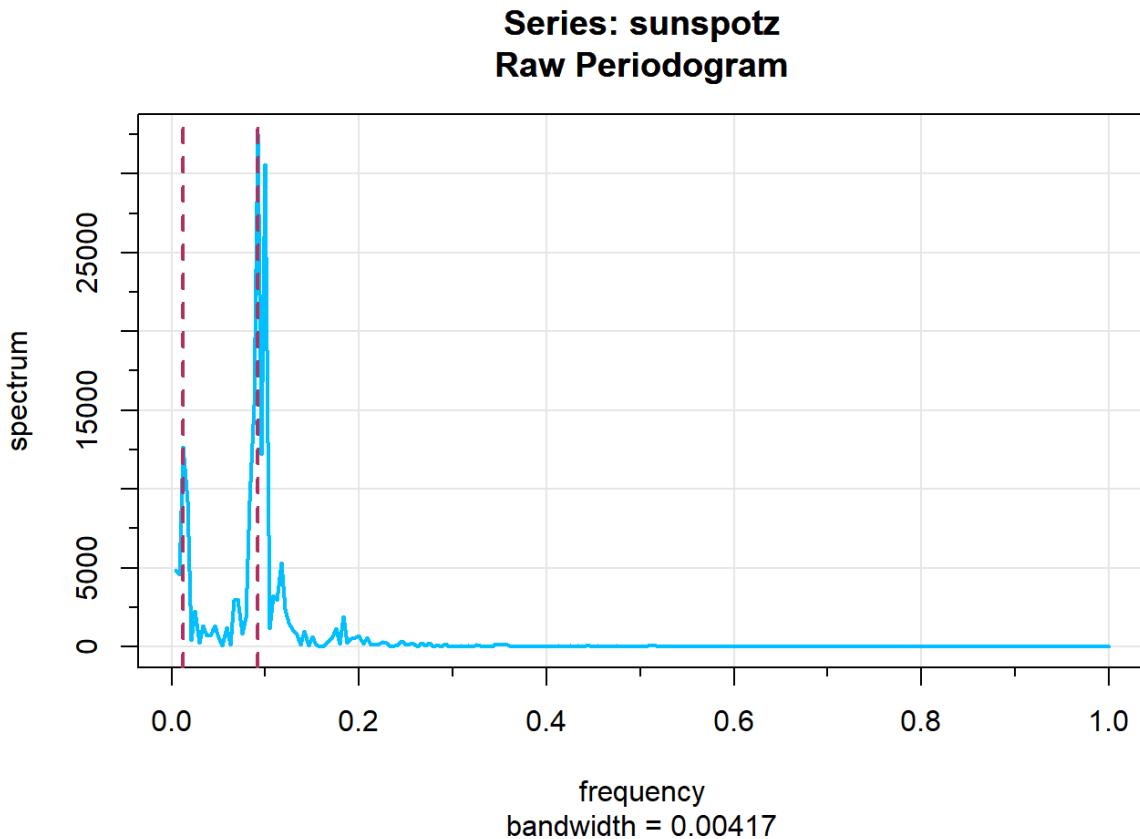
1st Diff and 12th Diff



7.1

Figure A.4 shows the biyearly smoothed (12-month moving average) number of sunspots from June 1749 to December 1978 with $n = 459$ points that were taken twice per year; the data are contained in `sunspotz`. With Example 7.4 as a guide, perform a periodogram analysis identifying the predominant periods and obtain confidence intervals. Interpret your findings.

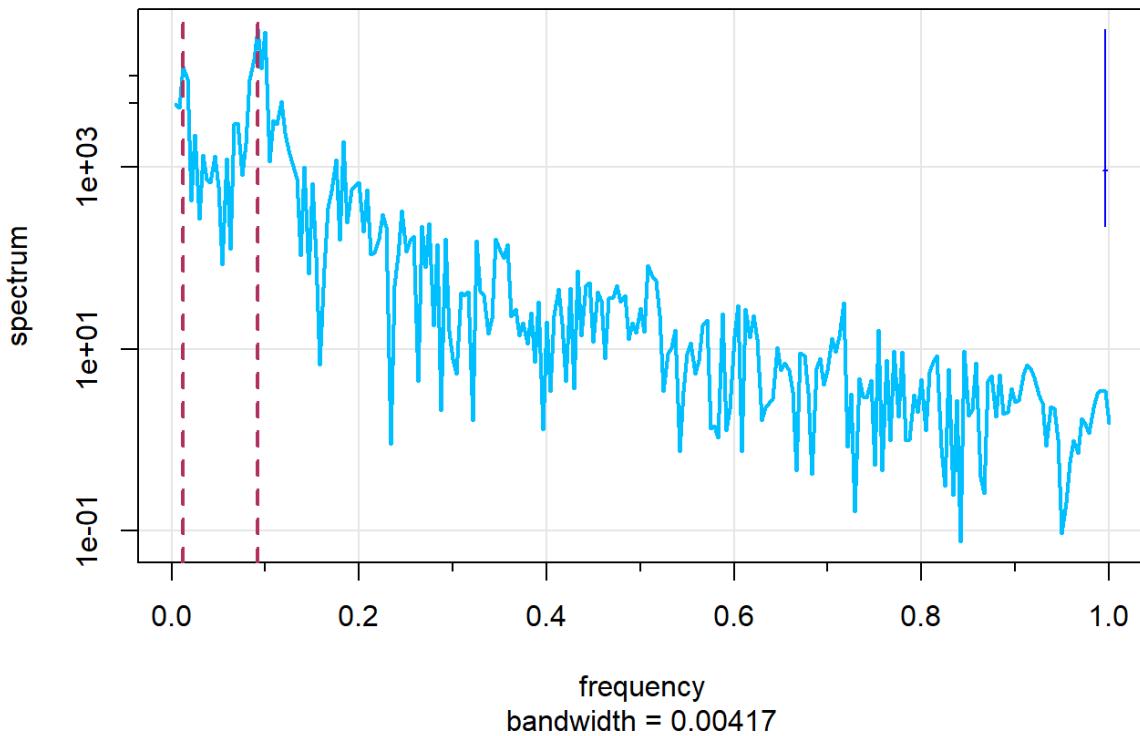
```
par(mfrow = c(1, 1))
mvspec(sunspotz, col = "deepskyblue1", lwd = 2)
abline(v = 1/10.9, lty = 2, lwd = 2, col = "maroon")
abline(v = 1/80, lty = 2, lwd = 2, col = "maroon")
```



```
# m = mvspec(sunspotz, col = "deepskyblue1", lwd = 2)
# m$details

# confidence intervals
mvspec(sunspotz, col = "deepskyblue1", lwd = 2, log = "yes")
abline(v = 1/10.9, lty = 2, lwd = 2, col = "maroon")
abline(v = 1/80, lty = 2, lwd = 2, col = "maroon")
```

Series: sunspotz Raw Periodogram



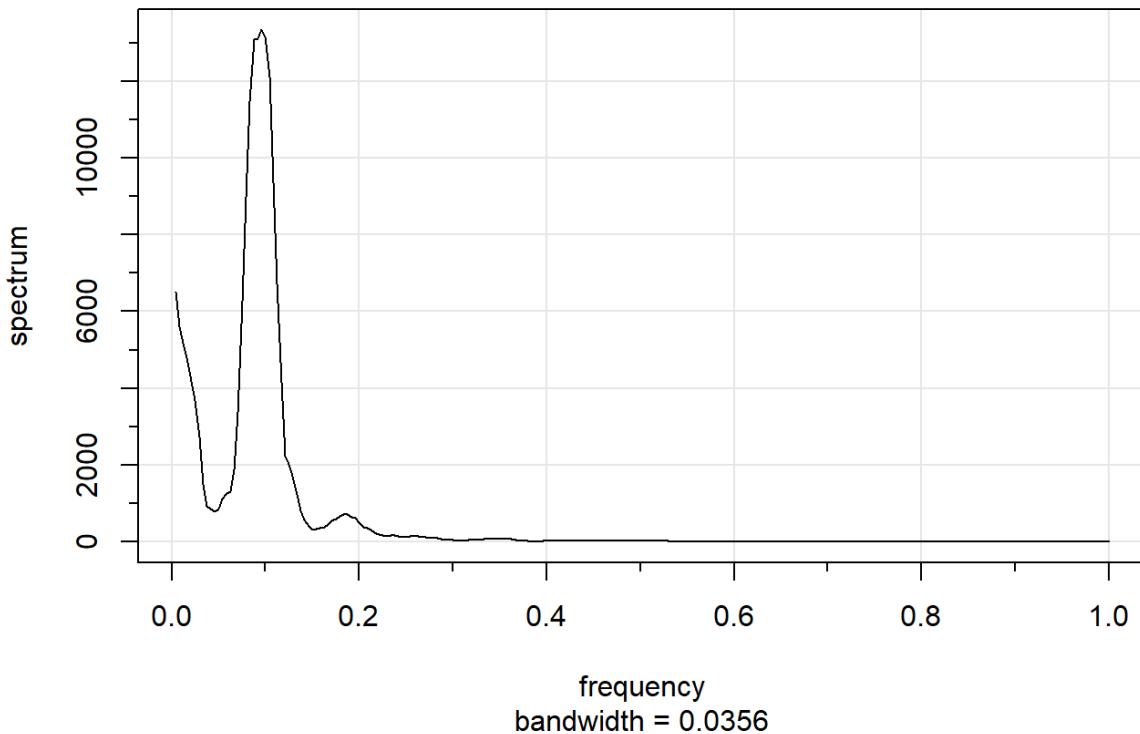
We can see from the “m\$details” output (not shown - super long) and the graph’s peaks that there are approximately 11-year (10.9 year) and 80-year cycles. We additionally see a 10-year cycle but it doesn’t show up as prominently in the logged data with the confidence interval. The confidence interval, displayed on the second plot, is on the far right and is quite wide. Because it is only based on the 2 degrees of freedom at each point, it is not of much use.

7.4

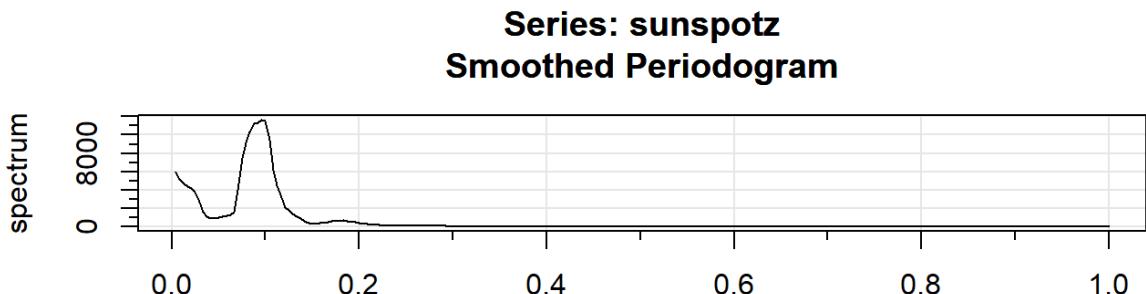
Repeat Problem 7.1 using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

```
mvspec(sunspotz, span = 8)
```

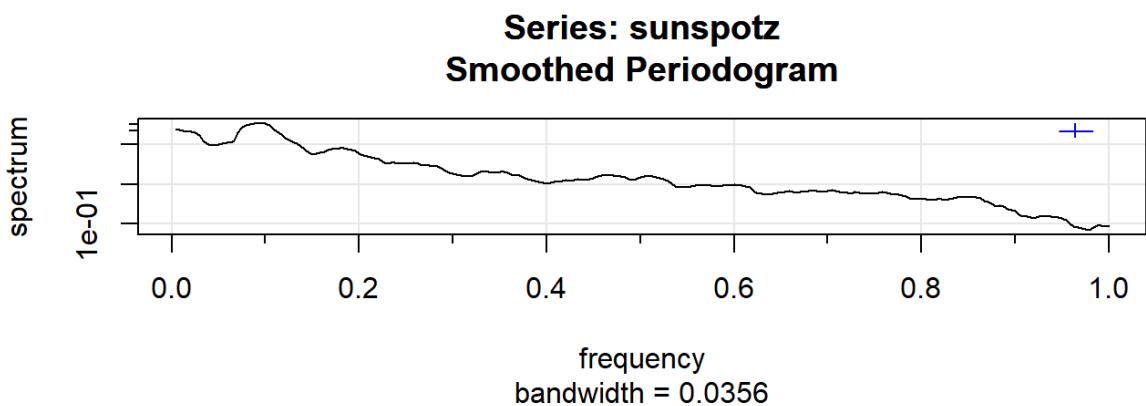
Series: sunspotz
Smoothed Periodogram



```
# n = mvspec(sunspotz, span = 8)
# n$Lh
par(mfrow = c(2, 1))
mvspec(sunspotz, spans = 8, taper = 0.2)
mvspec(sunspotz, spans = 8, taper = 0.2, log = 'y')
```



frequency
bandwidth = 0.0356



frequency
bandwidth = 0.0356

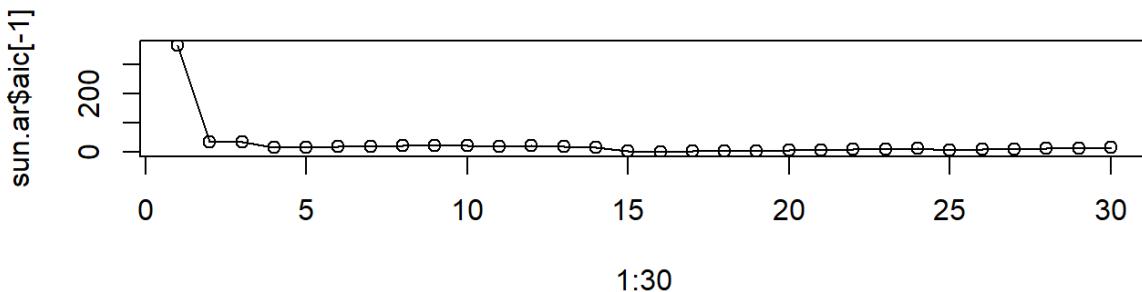
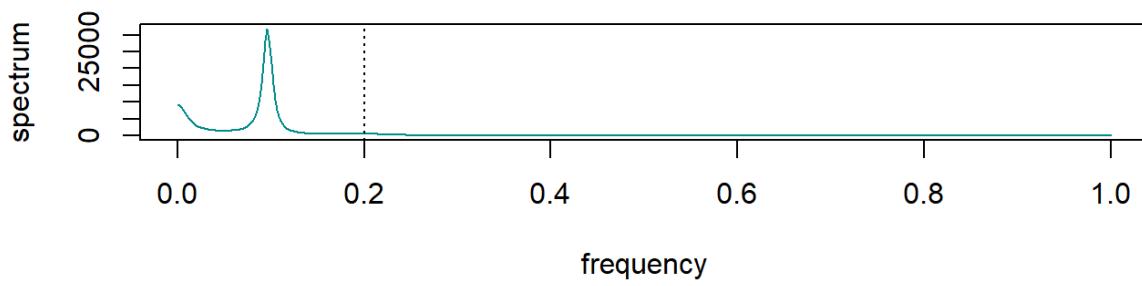
Once we smooth out the time series to see the peaks more clearly, we can add a taper. 20% is generally a good level of taper, and we see the result in the graph above. We don't want to go too far and lose the peaks, but the method of tapering helps keep those while also smoothing out the series. We can now see even more clearly that the peaks are close to 11, more accurately happening at about 10.43 years.

7.6

Often, the periodicities in the sunspot series are investigated by fitting an autoregressive spectrum of sufficiently high order. The main periodicity is often stated to be in the neighborhood of 11 years. Fit an autoregressive spectral estimator to the sunspot data using a model selection method of your choice. Compare the result with a conventional nonparametric spectral estimator found in Problem 7.4.

```
par(mfrow = c(2, 1))
spaic = spec.ar(sunspotz, log = "no", col = "cyan4")
abline(v = frequency(sunspotz)*1/10, lty = "dotted")
sun.ar = ar(sunspotz, order.max = 30)
plot(1:30, sun.ar$aic[-1], type = "o")
```

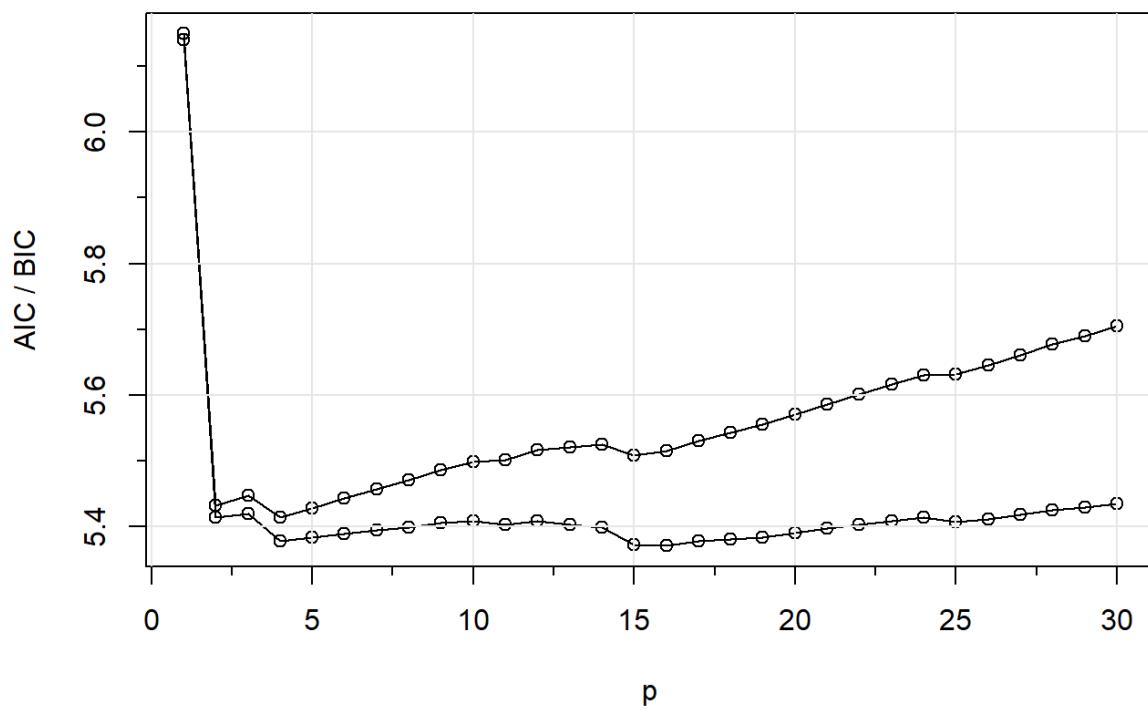
**Series: sunspotz
AR (16) spectrum**



```
spaic$method
```

```
## [1] "AR (16) spectrum "
```

```
par(mfrow = c(1, 1))
n = length(sunspotz)
c() -> AIC -> BIC
for (k in 1:30){
  sigma2 = ar(sunspotz, order = k, aic = FALSE)$var.pred
  BIC[k] = log(sigma2) + k*log(n)/n
  AIC[k] = log(sigma2) + (n+2*k)/n
}
IC = cbind(AIC, BIC+1)
ts.plot(IC, type="o", xlab = "p", ylab = "AIC / BIC")
Grid()
```



The `spec.ar` formula fits an AR(16) model on the sunspotz data. Since it relies on AIC and BIC for selection, it chooses the best model and then can be used to find the peaks associated with the sunspotz data. Instead of smoothing such as we did in 7.4, we rely on the chosen model.