Homework 9

Time Series Analysis (STAT 6391)

Willliam Ofosu Agyapong*

November 09, 2023

*woagyapong@miners.utep.edu, PhD Data Science, University of Texas at El Paso (UTEP).

Problem 6.1

We repeat the simulations and analyses in Example 6.1 and Example 6.2 in (Shumway and Stoffer 2019) with the following changes.

Part (a)

We changed the sample size to $n = 128$ and genereated and plotted the same series as in Example 6.1 in (Shumway and Stoffer 2019) such that

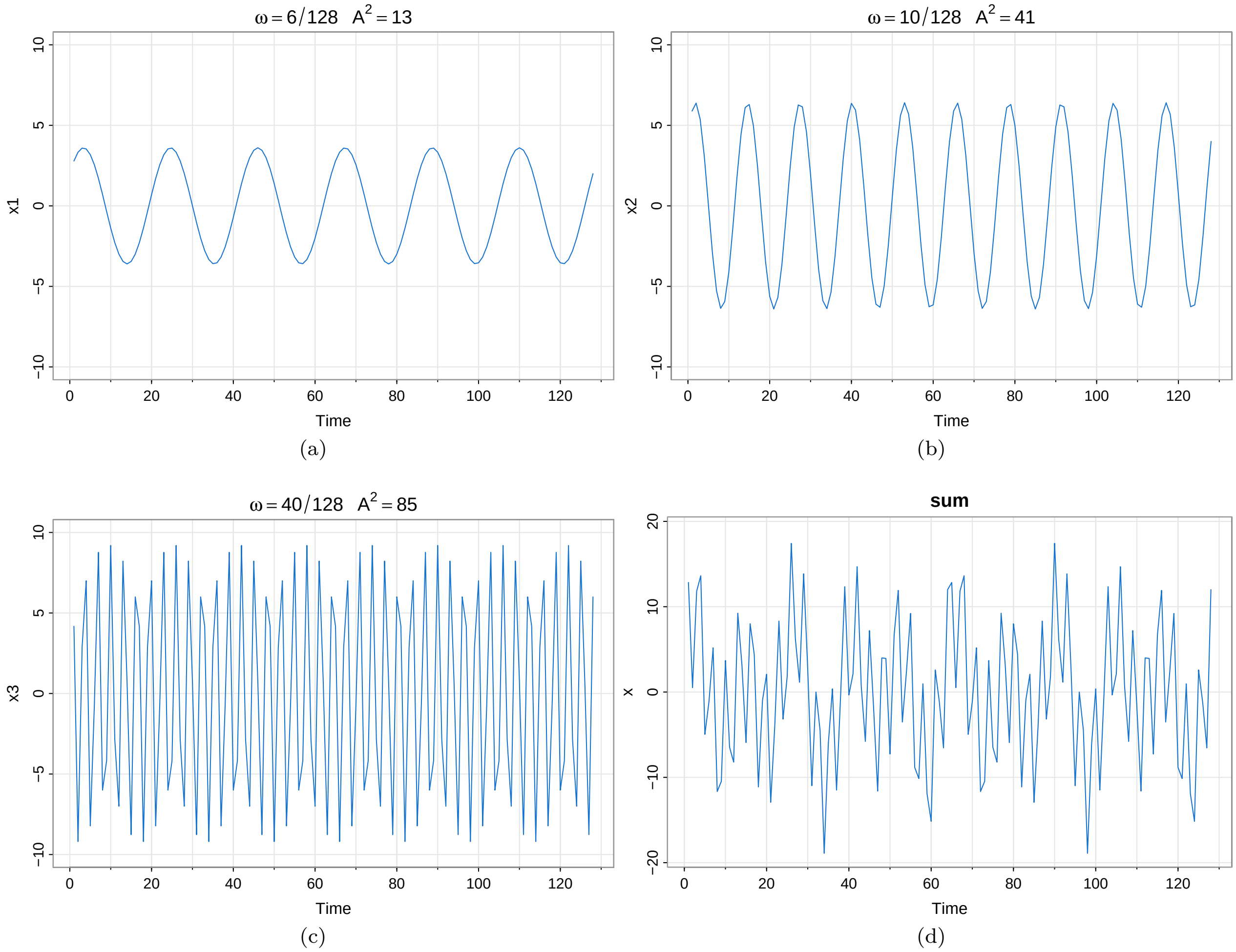
$$\begin{aligned}
x_{t1} &= 2cos\left(2\pi t\frac{6}{128}\right) + 3sin\left(2\pi t\frac{6}{128}\right), \\
x_{t2} &= 4cos\left(2\pi t\frac{10}{128}\right) + 5sin\left(2\pi t\frac{10}{128}\right), \\
x_{t1} &= 6cos\left(2\pi t\frac{40}{128}\right) + 7sin\left(2\pi t\frac{40}{128}\right),
\end{aligned}$$

and

$$x_t = x_{t1} + x_{t2} + x_{t3},$$

for $t = 1, \ldots, 128$.

The resulting series are exhibited in Figure 1 along with the corresponding frequencies and squared amplitudes (which gives the maximum and minimum values that the series can attain).



(a) (b) (c) (d)

Figure 1: Periodic components and their sum as described in the above series generating equations.

Comparing the results here to the ones in Example 6.1 of (Shumway and Stoffer 2019), the major difference between the series is the change in the **fundamental frequencies** ($\omega = j/n$), since they are functions of the sample size, n. Given the same constant, j, then as sample size increases, the fundamental frequency will decrease, leading to a high period in the series (where period $= 1/\omega$). For instance, for the series x_1, when $n = 100$, $\omega = 0.06 \Rightarrow$ period $= 1/0.06 \approx 16$ points, but when $n = 128$, $\omega = 0.0469 \Rightarrow$ period $= 1/0.0469 \approx 21$ points. Similarly, for the series x_3, when $n = 100$, $\omega = 0.40 \Rightarrow$ period $= 1/0.40 \approx 2$ points, however when $n = 128$, $\omega = 0.3125 \Rightarrow$ period $= 1/0.3125 \approx 3$ points.

Part (b)

Similarly, we computed and plotted the periodogram of the series, x_t, generated in part (a) as was done in Example 6.2 of (Shumway and Stoffer 2019).

As expected, mirroring effect at the folding frequency of .5 is apparent. Despite the change in sample size, the scaled periodogram of the data, x_t with $n = 128$ undoubtedly identifies the three (3) components x_{t1}, x_{t2}, and x_{t3} of x_t. The scaled periodogram values corresponding to the 3 components are 13, 41, and 85, which are exactly the squared amplitudes of the components as before.

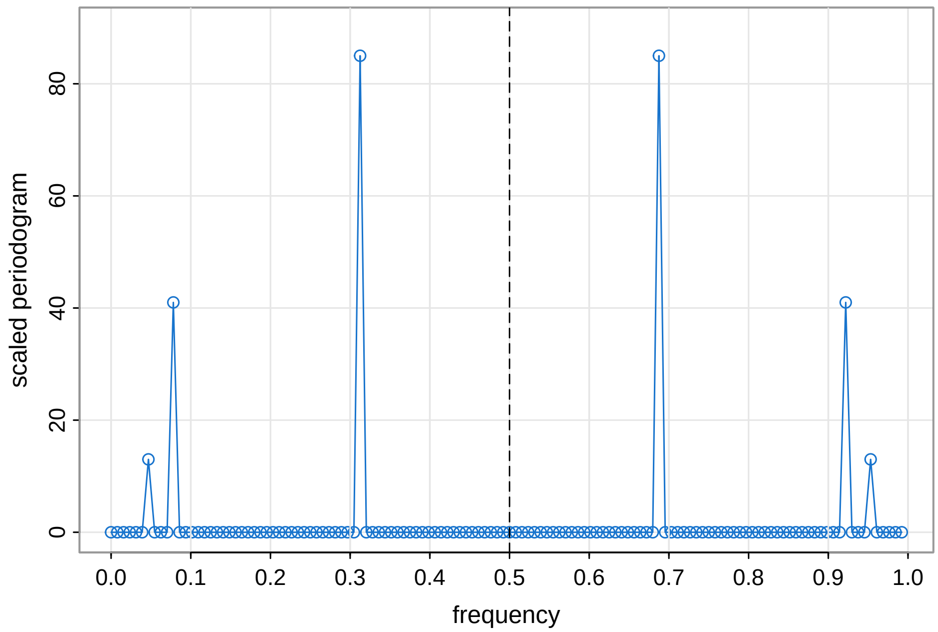


Figure 2: The scaled periodogram of the data generated in part (a) of problem 6.1.

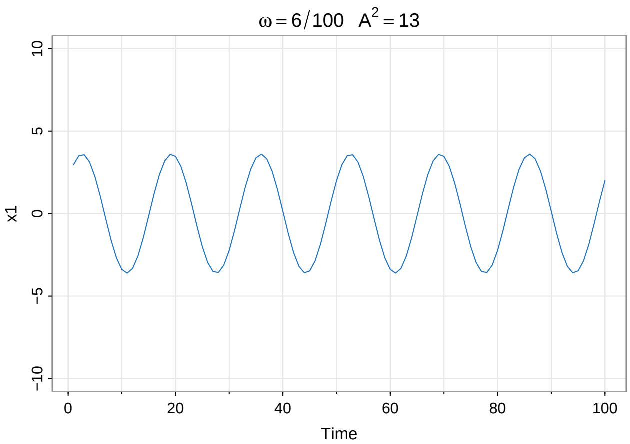
Part (c)

In this part of the problem, we repeated the analyses of part(a) and part (b) but with $n = 100$ (as in Example 6.1 of (Shumway and Stoffer 2019)), and added noise, $w_t \sim iid\ N(0, \sigma_w = 5)$, such that

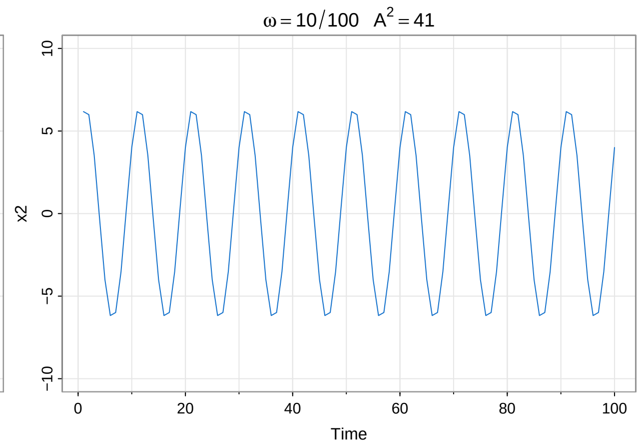
$$x_t = x_{t1} + x_{t2} + x_{t3} + w_t.$$

That is, we simulated and plotted the data, and then plotted the periodogram of x_t as shown in Figure 3 and Figure 4, respectively.

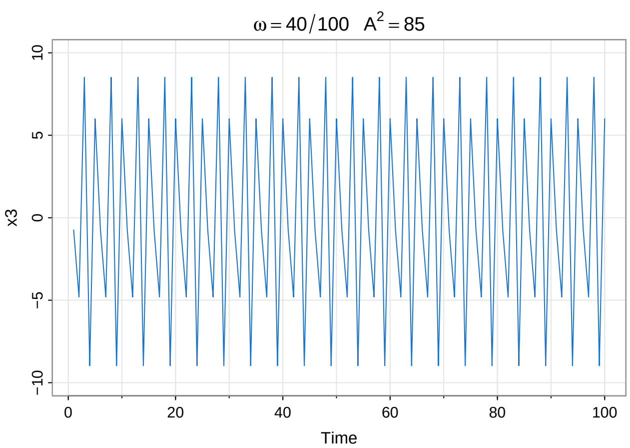
The periodogram of the new x_t with noise added also identifies 3 dominant frequencies obviously corresponding to the 3 components in the data. Here, the heights of the dominant scaled periodogram are 18.22, 33.30 and 79.89, which are not exactly the same as the squared amplitudes. Therefore, we learn that, even though adding noise to the series seem to change the magnitude of the dominant frequencies, the scaled periodogram can still identify the important underlying components.



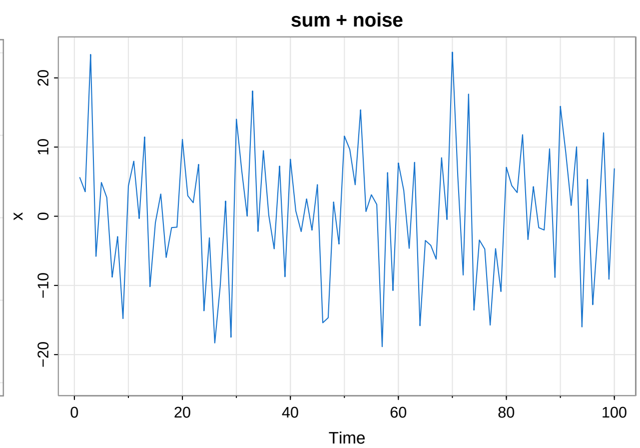
(a)



(b)



(c)



(d)

Figure 3: Periodic components and their sum as described in the above series generating equations but with iid normally distributed noise added to x_t.

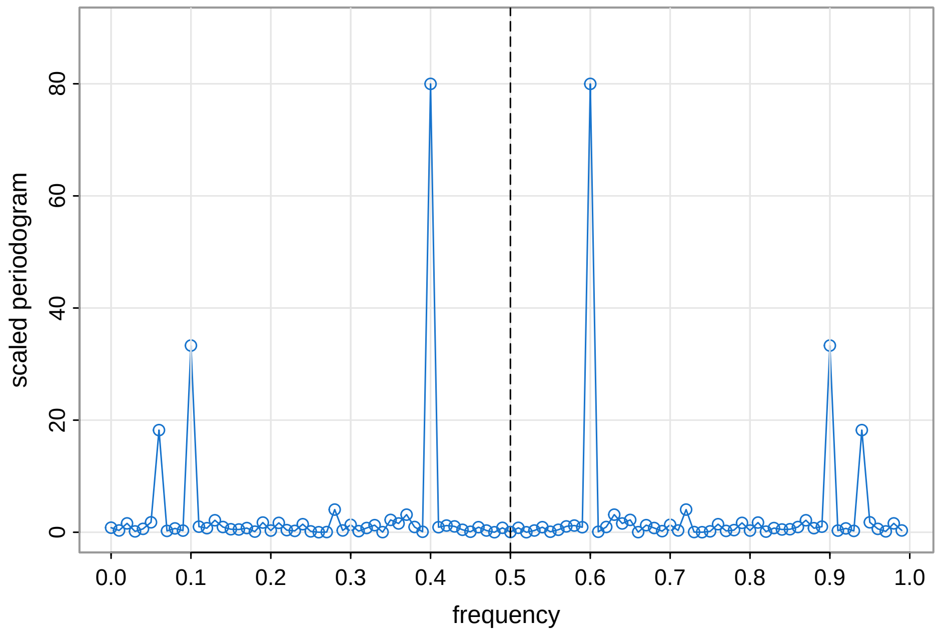
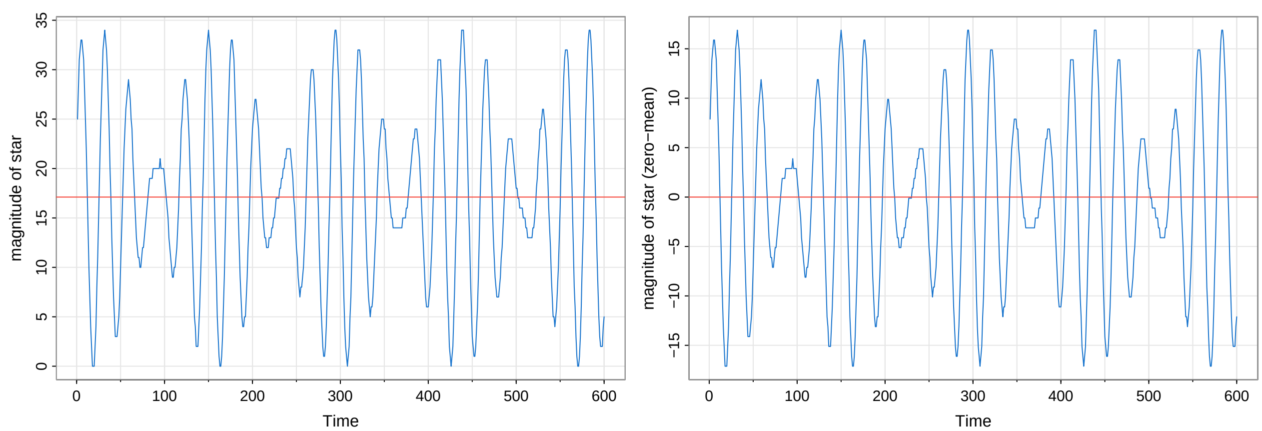


Figure 4: The scaled periodogram of the data generated with the addition of iid normally distributed noise.

Problem 6.3

To remove the mean of the series (as hinted out), we simply subtracted the mean of the actual `star` series from the series itself. Time series plots of the actual series and the zero-mean series are displayed in Figure 5. The red horizontal lines represent the means of the two series, which confirms that we succeeded in removing the mean from the series.

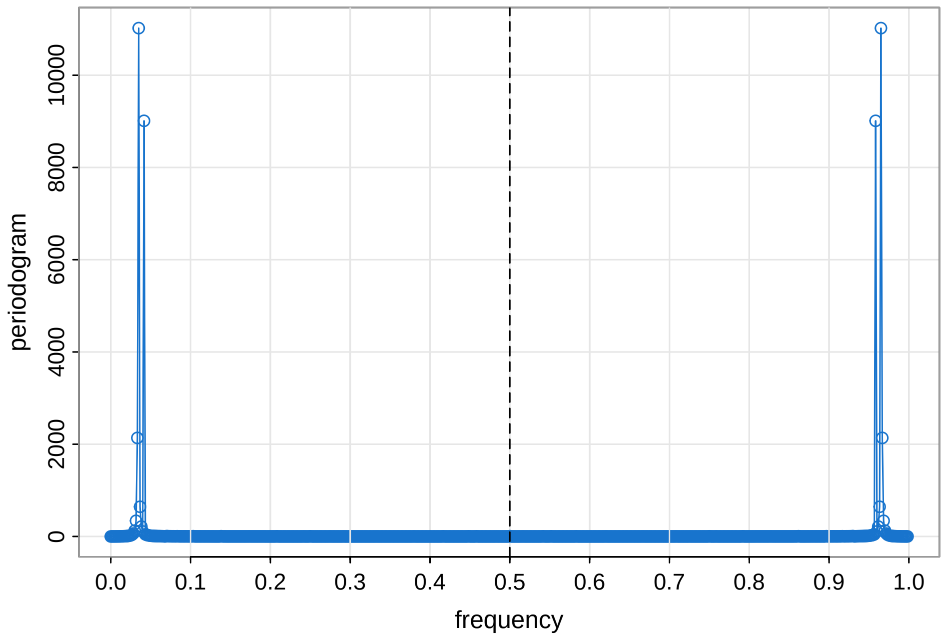
The periodogram analysis was then performed on the zero-mean data shown in plot (b) in Figure 5. Plots of the computed periodogram and scaled periodogram are presented in Figure 6, which shows three (3) prominent periodic components of the data clustered around very low frequencies. The heights of the scaled periodogram for the three prominent components are 14.25, 60.07, and 73.47.



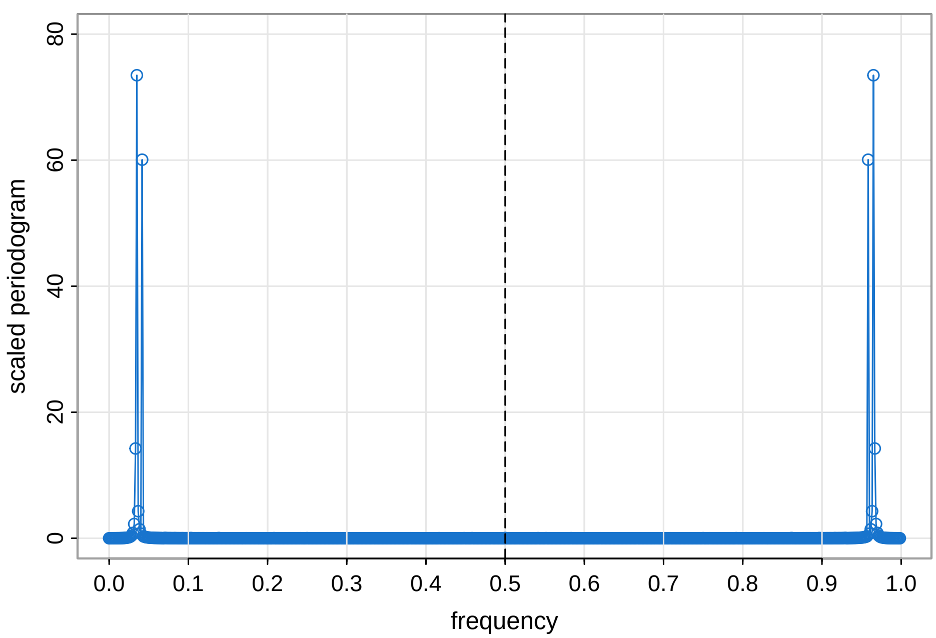
(a) Actual series (mean is about 17)

(b) Zero-mean series

Figure 5: The magnitude of star series data. Horizontal red line represents the mean of the series.



(a) Periodogram



(b) Scaled periodogram

Figure 6: The periodogram and scaled periodogram of the magnitude of star data.

Appendix R Codes for the Analysis

```
# Set global options for output rendering
knitr::opts_chunk$set(eval = T, echo = F, warning = F, message = F,
                      fig.pos = "H", out.extra = "", fig.align = "center",
                      cache = F, comment="")

#----------------- Load required packages
# library(dplyr)
library(knitr)
library(kableExtra)
# library(broom)
library(stats)
library(astsa)


#----------------- set the current working directory to the file path
setwd(dirname(rstudioapi::getSourceEditorContext()$path))



#---------- Problem 6.1

# generate individual series
n <- 128; t <- 1:n
x1 <- 2*cos(2*pi*t*6/n) + 3*sin(2*pi*t*6/n)
x2 <- 4*cos(2*pi*t*10/n) + 5*sin(2*pi*t*10/n)
x3 <- 6*cos(2*pi*t*40/n) + 7*sin(2*pi*t*40/n)

# construct xt
x <- x1+x2+x3

# plot the series
tsplot(x1, ylim=c(-10,10), col=4,main=expression(omega==6/128~~~A^2==13))
tsplot(x2, ylim=c(-10,10), col=4,main=expression(omega==10/128~~~A^2==41))
tsplot(x3, ylim=c(-10,10), col=4,main=expression(omega==40/128~~~A^2==85))
tsplot(x,  ylim=c(-19,19), col=4,main="sum")


# calculating scaled periodogram
P  = Mod(fft(x)/sqrt(n))^2  # periodogram
sP = (4/n)*P                # scaled peridogram
Fr = (0:(n-1))/n            # fundamental frequencies
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))
# generate individual series
n <- 100; t <- 1:n
x1 <- 2*cos(2*pi*t*6/n) + 3*sin(2*pi*t*6/n)
x2 <- 4*cos(2*pi*t*10/n) + 5*sin(2*pi*t*10/n)
x3 <- 6*cos(2*pi*t*40/n) + 7*sin(2*pi*t*40/n)
```

```
# construct xt
set.seed(123)
x <- x1+x2+x3 + rnorm(n, sd=5)

# plot the series
tsplot(x1, ylim=c(-10,10), col=4,main=expression(omega==6/100~~~A^2==13))
tsplot(x2, ylim=c(-10,10), col=4,main=expression(omega==10/100~~~A^2==41))
tsplot(x3, ylim=c(-10,10), col=4,main=expression(omega==40/100~~~A^2==85))
tsplot(x,  ylim=c(-24,24), col=4,main="sum + noise")
n <- 100
P  <- Mod(fft(x)/sqrt(n))^2  # periodogram
sP <- (4/n)*P               # scaled peridogram
Fr <- (0:(n-1))/n           # fundamental frequencies
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))

#---------- Problem 6.3

# plotting the series
x <- star
x_diff <- x - mean(x)
tsplot(x, col=4,ylab = "magnitude of star")
abline(h=mean(x), lty=1, col=2)
tsplot(x_diff, col=4,ylab = "magnitude of star (zero-mean)")
abline(h=mean(x_diff), lty=1, col=2)


# computing periodogram
n <- length(x_diff)
P  <- Mod(fft(x_diff)/sqrt(n))^2  # periodogram
sP <- (4/n)*P                    # scaled peridogram
Fr <- (0:(n-1))/n                # fundamental frequencies
tsplot(Fr, P, type="o", xlab="frequency", ylab="periodogram",
       col=4, ylim=c(0, max(P)+5))
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))


# scaled periodogram
tsplot(Fr, sP, type="o", xlab="frequency", ylab="scaled periodogram",
       col=4, ylim=c(0,80))
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))

# find the prominent frequencies:
large_sP <- sort(unique(round(sP,2)), decreasing = T)[1:5]
```

References

Shumway, Robert, and David Stoffer. 2019. *Time Series: A Data Analysis Approach Using r*. CRC Press.