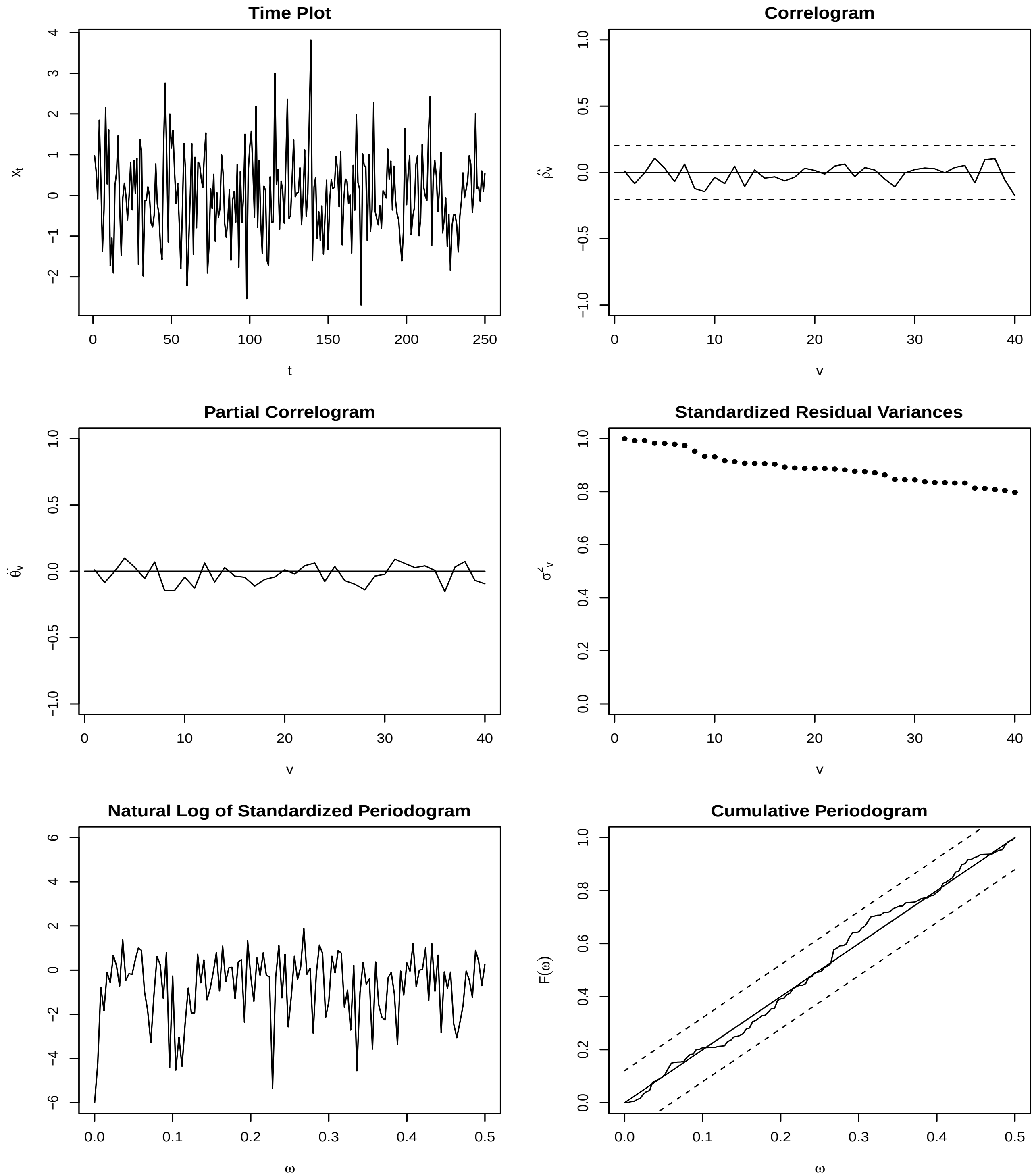
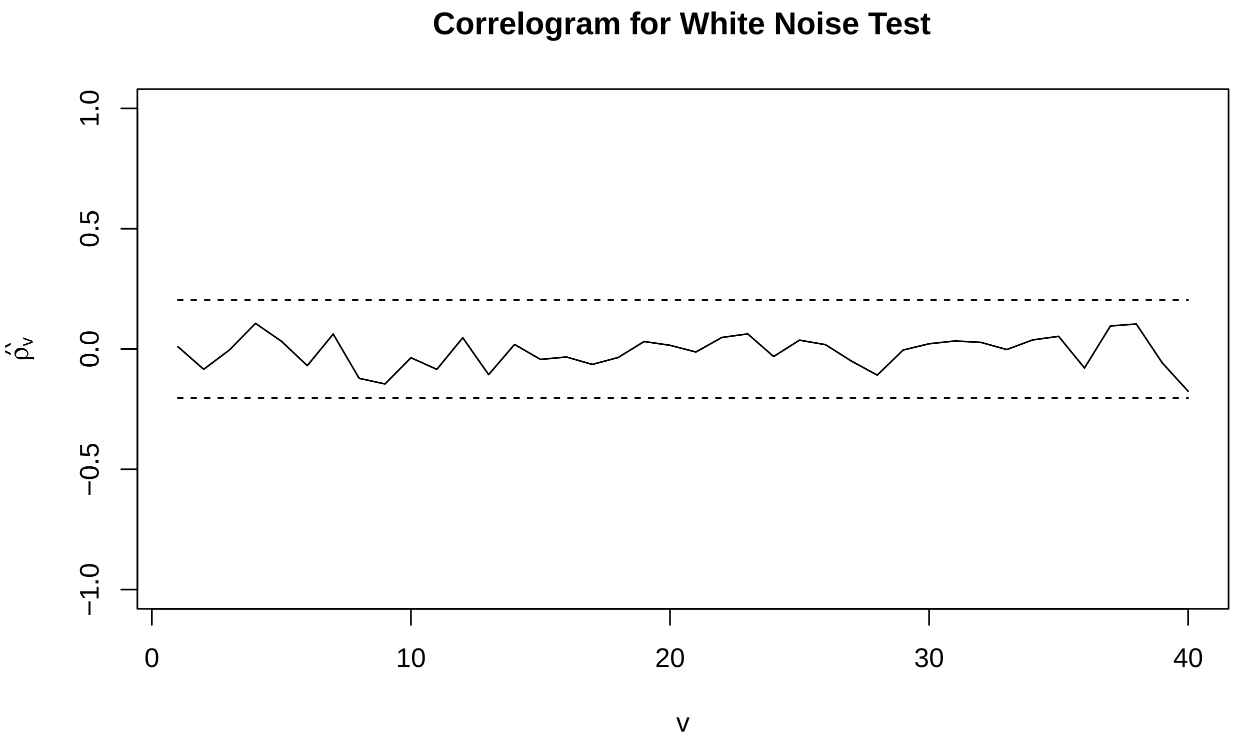
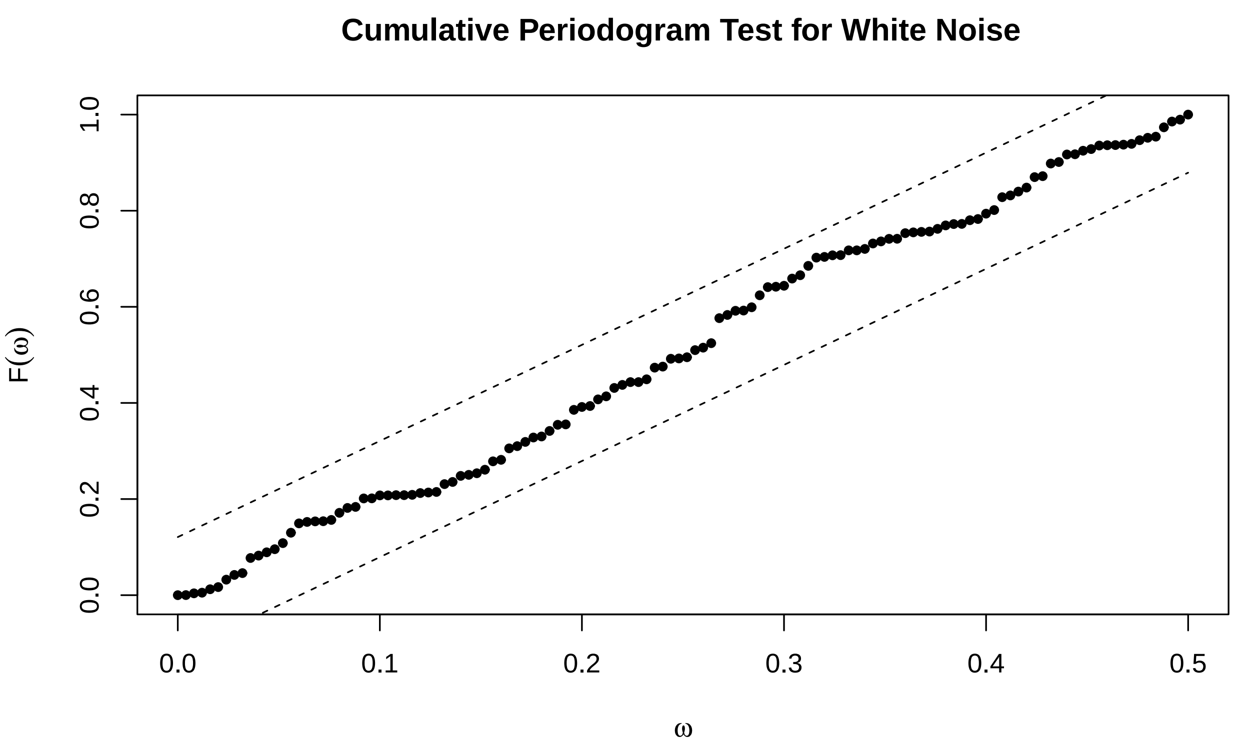
Series1 HW3 Alex Salo: Time Series Description Using Chapter 2 Tool Set



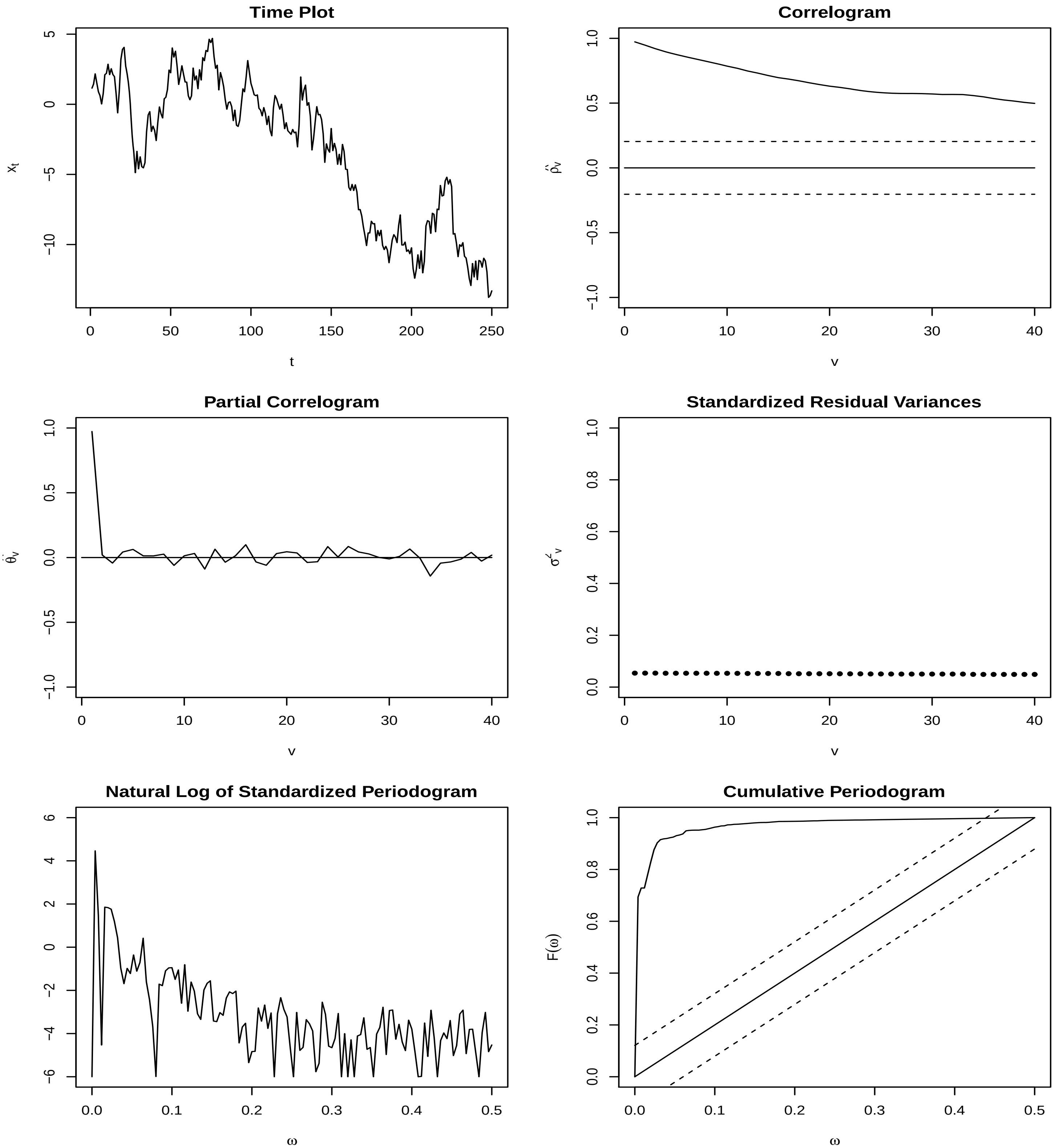
Looks exactly like the **normal white noise**. Correlogram fits within the significance threshold (low correlation). The same with partial. Residual variance close to one – we don't explain variance at all. Periodogram doesn't show any distinct frequencies. Cumulative periodogram stays within Chi^2 critical values for hypothesis test of **normal white noise.**



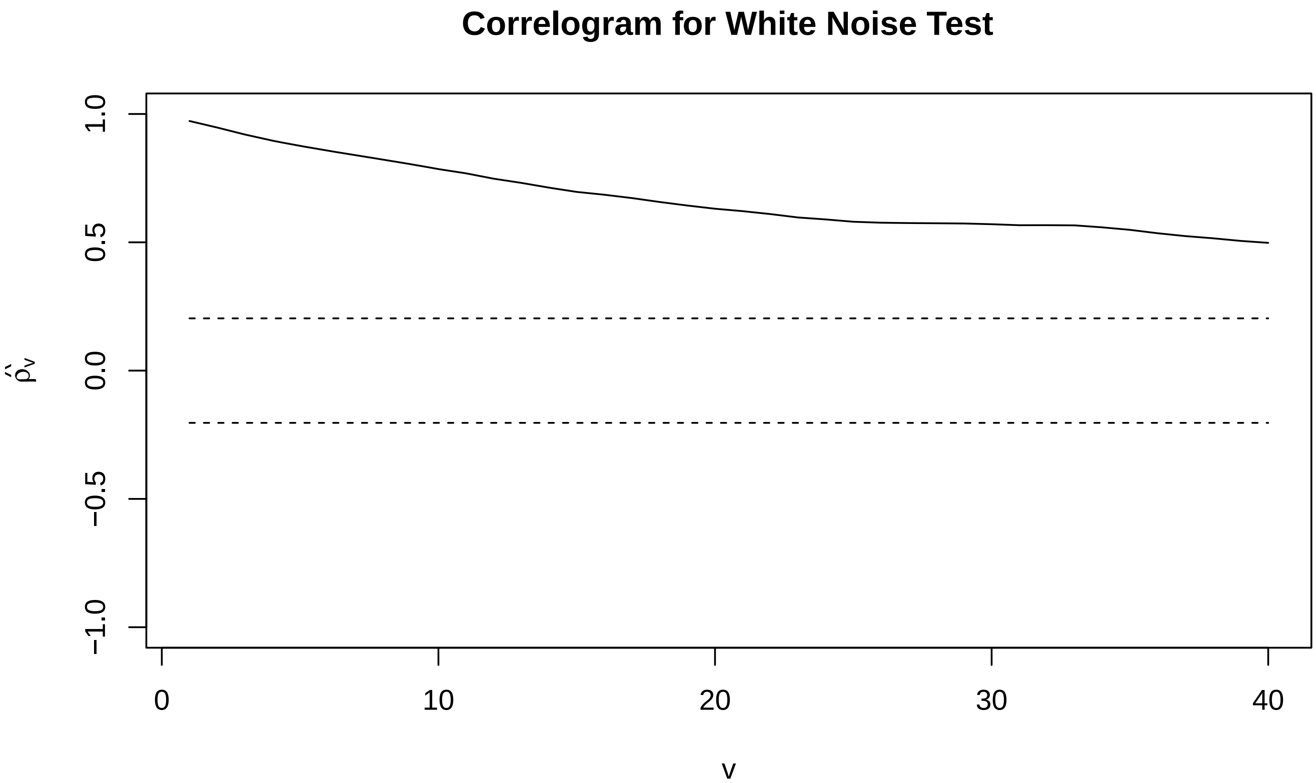


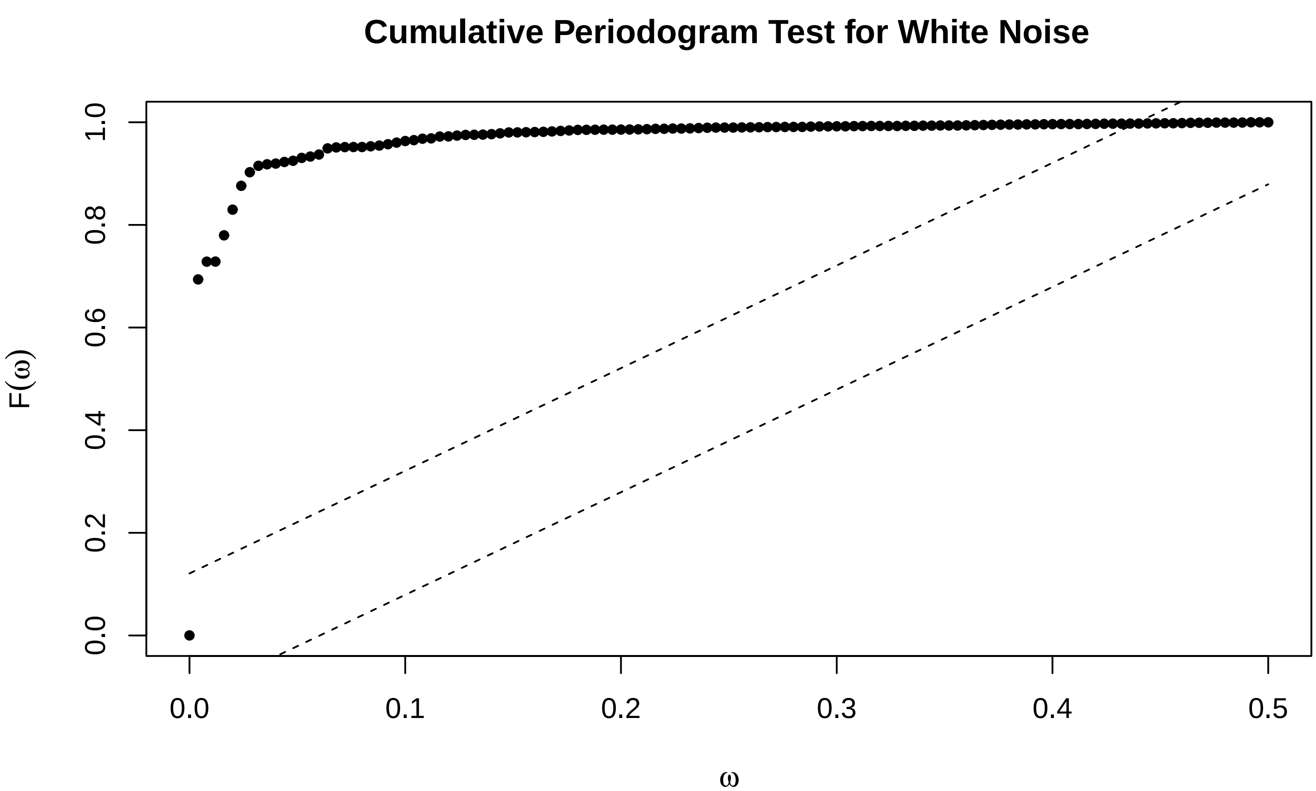
According to this plots which show the results of hypothesis testing (H0 = TS is a White Noise) we can rule out that this TS is indeed a normal white noise.

Series2



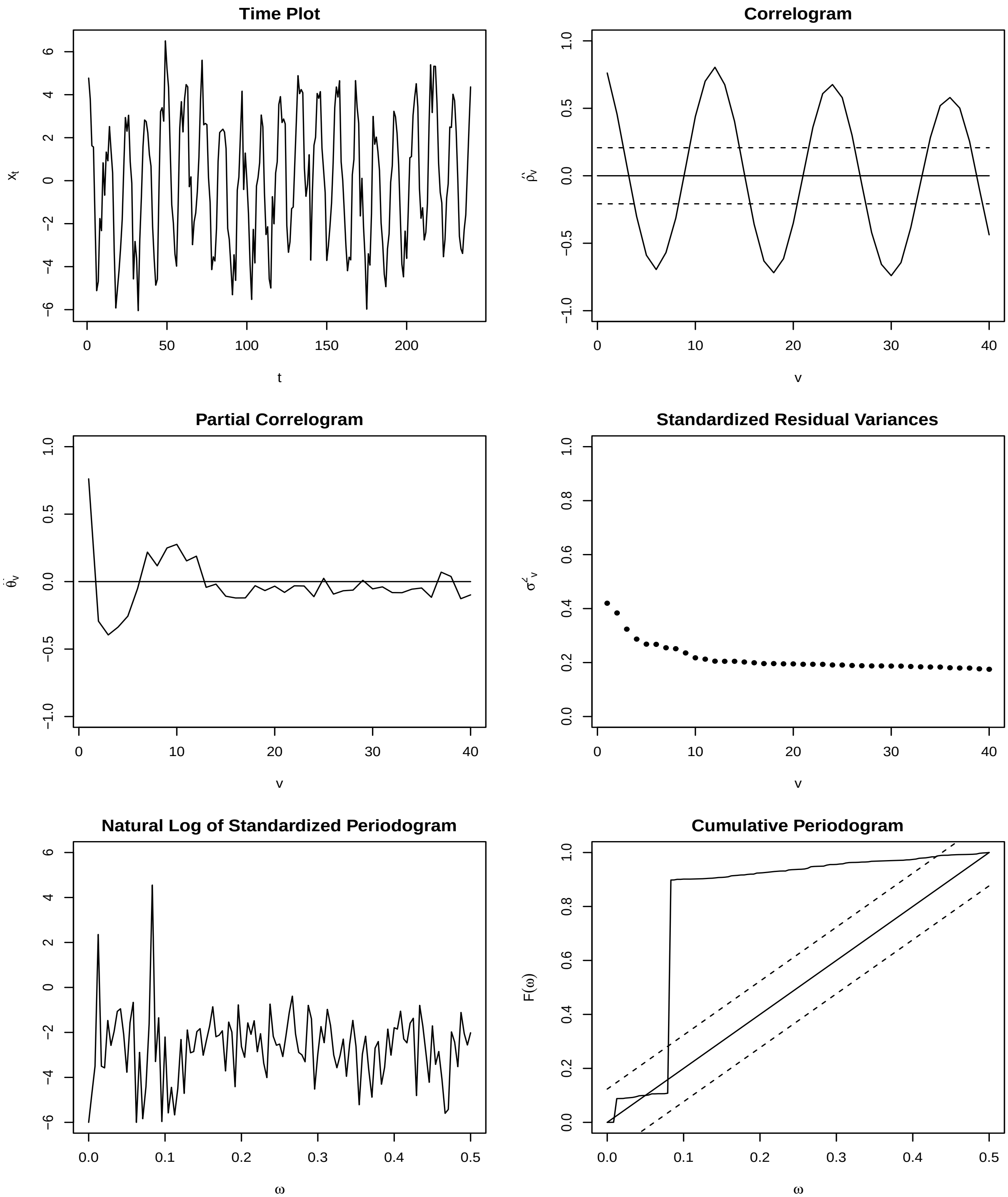
Does not look like WN. Guess would be Random Walk (RW). Corelogram shows gradual fade with the increase of lag. Partial shows that practically all the correlation is coming from the **lag=1** and other lags just linked through that – supports RW. Residuals are practically nonexistent meaning we explain everything. Periodogram shows the prevalence of the low frequencies which again supports RW. Cumulative periodogram looks exactly how the RW's should look like. No doubt – this is the **Random Walk series** with the model $x_t = x_{t-1} + e_t$**.**





Clearly we **reject H0** about this series being a white noise.

Series3



Looks like a mix of pure sinusoids. Corelogram is sinusoid – supports the previous hypothesis. Partial shows that after lag=15 there is nothing additional to the correlation – agrees with sine version. Periodogram makes it very easy – we can even state that we have two pure sinusoids (with maybe some not significant noise) with the frequencies (240 * 0.02 and 250 * 0.08) ~ 1/4, 1/16 (for N = 250). Cumulative periodogram supports this version since there are two distinct raises in the function.

Thus we can claim that this time series is the **sum of two sinusoids with white noise.**

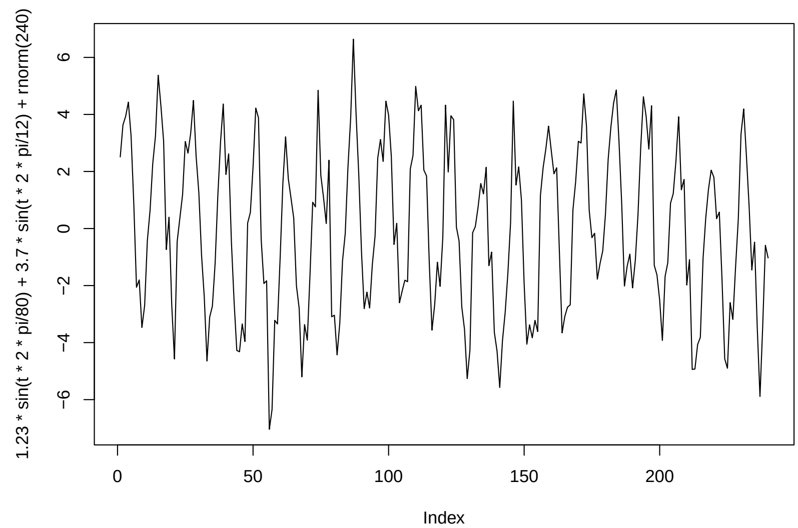
To be precise, and since we actually have the data, we can get exact values of those to sinusoids:

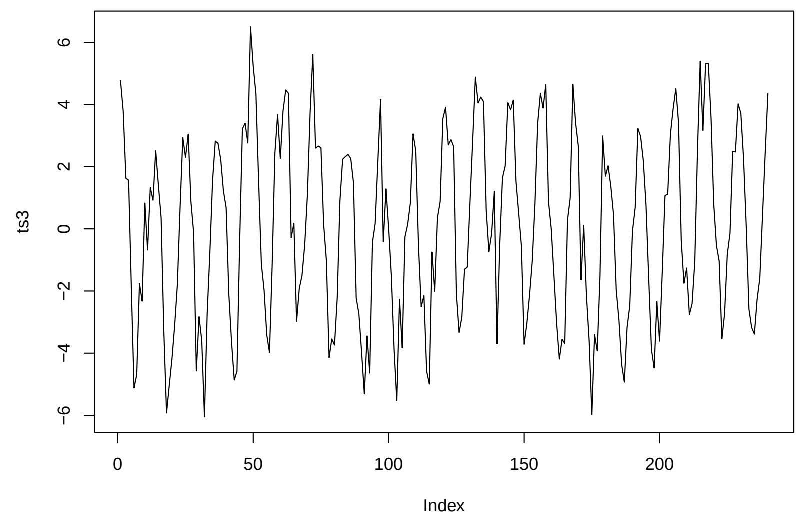
```
ts3 = read.table('series3.dat')$V1
plotsp(perdgm(ts3), length(ts3), sd(perdgm(ts3)))
ts3.freqs =
round(Mod(fft(ts3))[1:(length(ts3)/2+1)] * 2 /
length(ts3), 6)
ts3.freqs[ts3.freqs < 1] = 0

amps = ts3.freqs[ts3.freqs > 0]
>[1] 1.2 3.8
freqs = length(ts3) / (-1 + which(ts3.freqs > 0))
> [1] 80 12

par(mfrow=c(2,1))
t = 1:240
plot(1.23*sin(t*2*pi/80) + 3.7*sin(t*2*pi/12) +
rnorm(240), type="l")
plot(ts3, type="l")
```

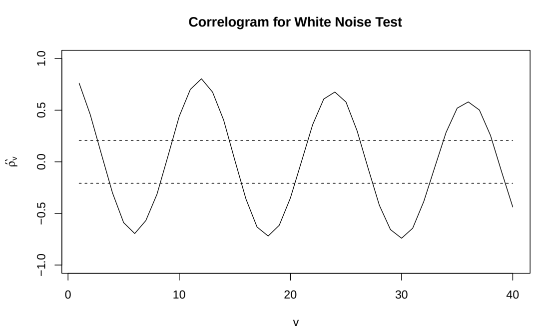
Thus freqs are 80 and 12, and amps are 1.2 and 3.8 respectively; plus normal white noise with zero mean and variance of one.

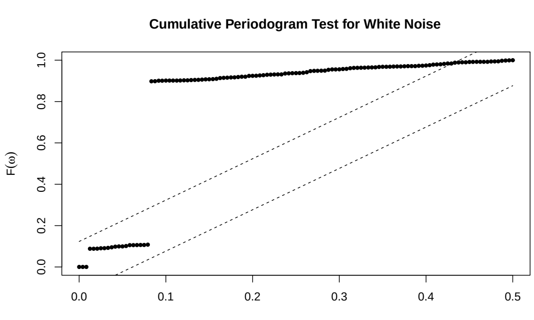




We conclude that the process generated the series looks like:

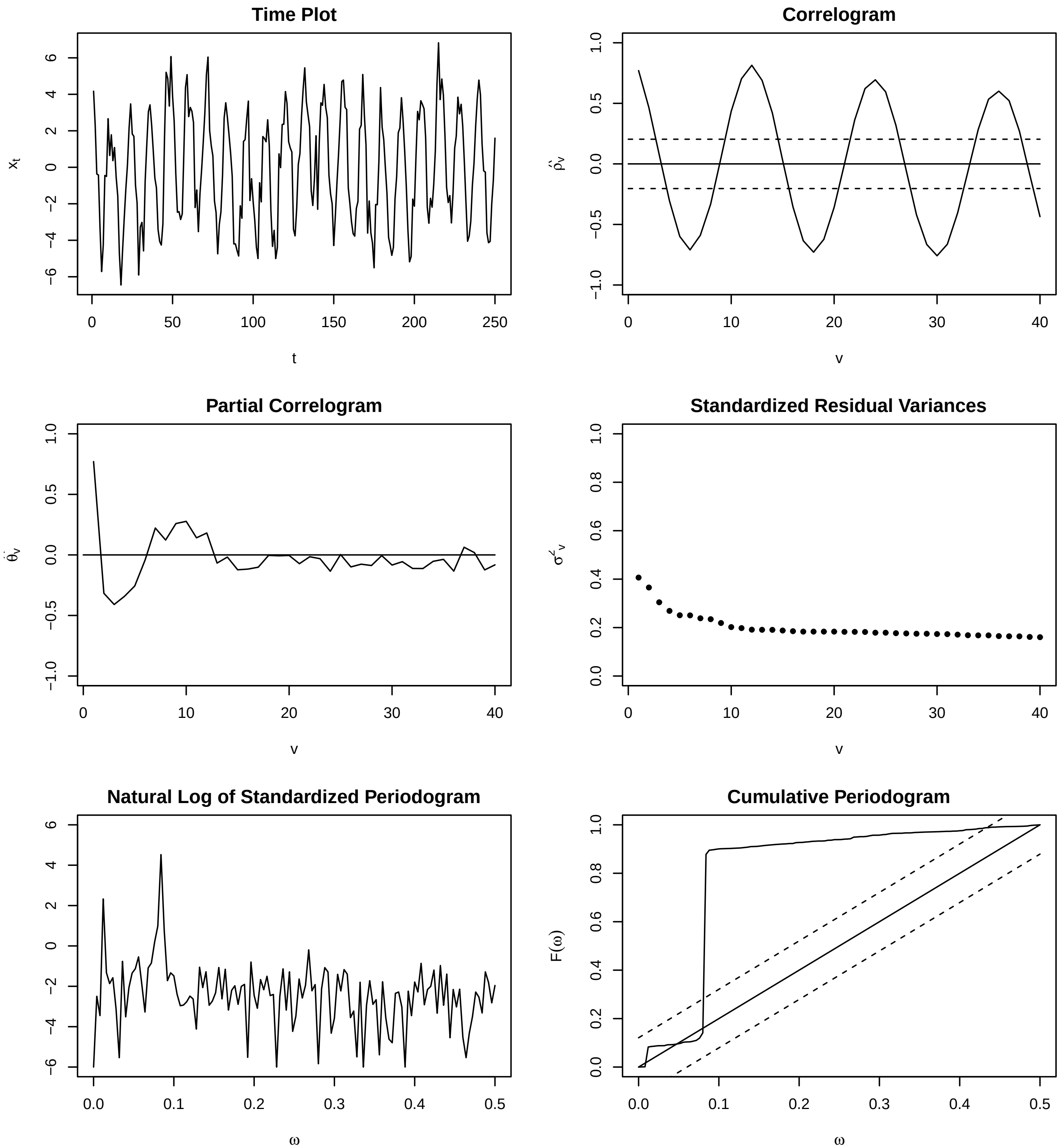
$$x_t = 1.2cos\frac{2\pi(t-1)}{80} + 3.8cos\frac{2\pi(t-1)}{12}$$





This clearly shows that we **reject H0** about TS being a white noise.

Series4



This time series looks almost exactly like the previous one. However Cumulative plot has one distinct feature: two increases are not so sharp as in the TS3. Also peaks on natural log periodogram are not so distinct. That means that we were unable to get the natural frequencies, in other words – **frequencies are not of the form $w_n = (k-1)/n$**.

Indeed, want we try to restore the frequencies as we did before:

```
ts4.freqs = round(Mod(fft(ts4))[1:(length(ts4)/2+1)] * 2 / length(ts4), 6)
ts3.freqs[ts4.freqs < 1] = 0
ts4.freqs[ts4.freqs > 0] # amps
length(ts4) / (-1 + which(ts4.freqs > 0)) #freqs
```

```
>[1] 0.112927 0.121089 0.167063 1.186650 0.172165 0.100908 … [250] 0.080723
```

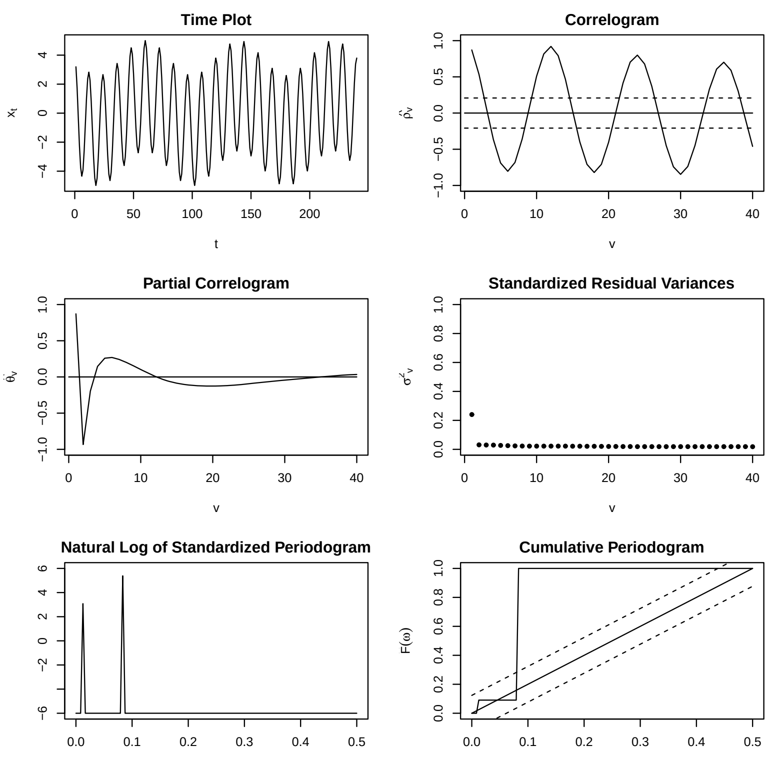
We see that there are no distinct frequencies. However we can judge that the model was exatly the same as the previous one:

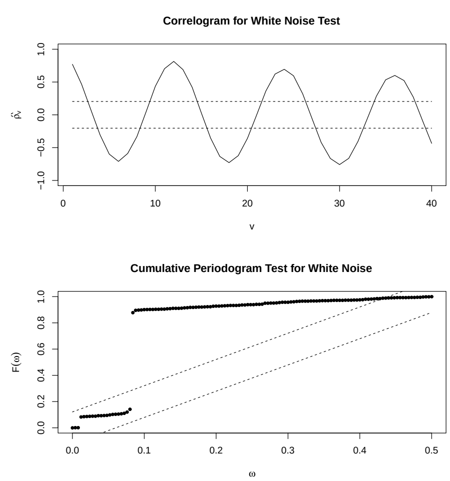
$$x_t = 1.2cos\frac{2\pi(t-1)}{80} + 3.8cos\frac{2\pi(t-1)}{12}$$

To double check:

```
ts4 = ts4[1:240] - ts1[1:240] #remove wn and use only 240 data points
descplot(ts4, 40)
```

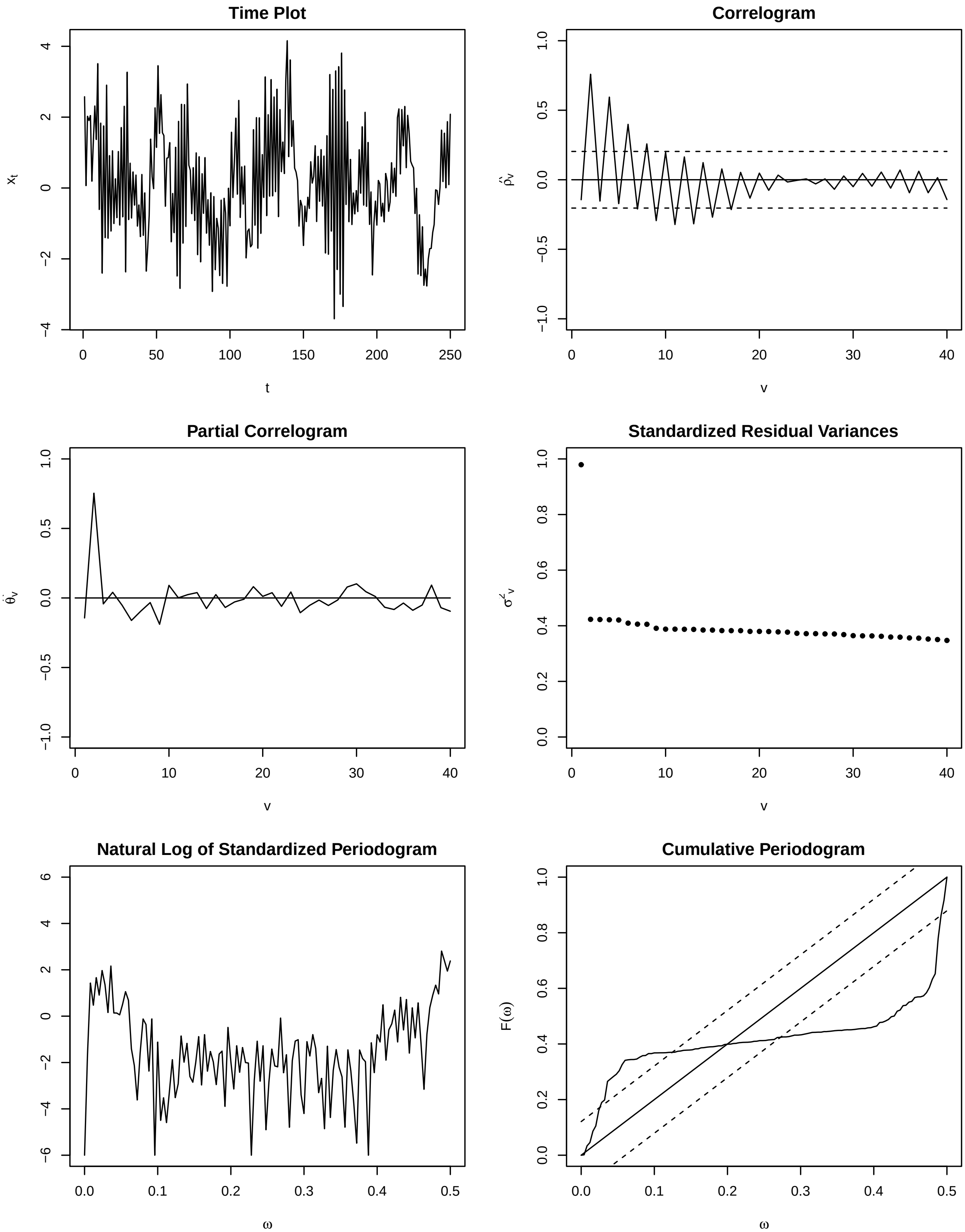
Then we see exactly the same picture as in TS3. Thus additional 10 observations hindered our analysis since 250 was not a common divisor of the process' frequencies 1/80 and 1/12.





Clearly not a white noise.

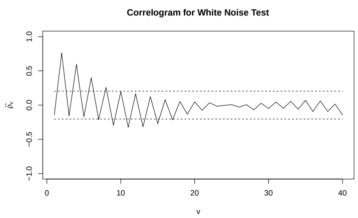
Series5

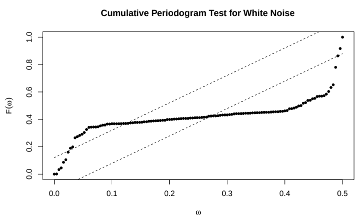


This time series looks like a typical **Autoregressive Process**. Correlogram in particular demonstrates that. Partial suggest that most (or all) correlation is coming from the lag=2. Thus the model should like something like $\mathbf{x_t=0.7x_{t-2} + e_t}$

Periodograms support this hypothesis.

Naturally, white noise hypothesis does not hold:

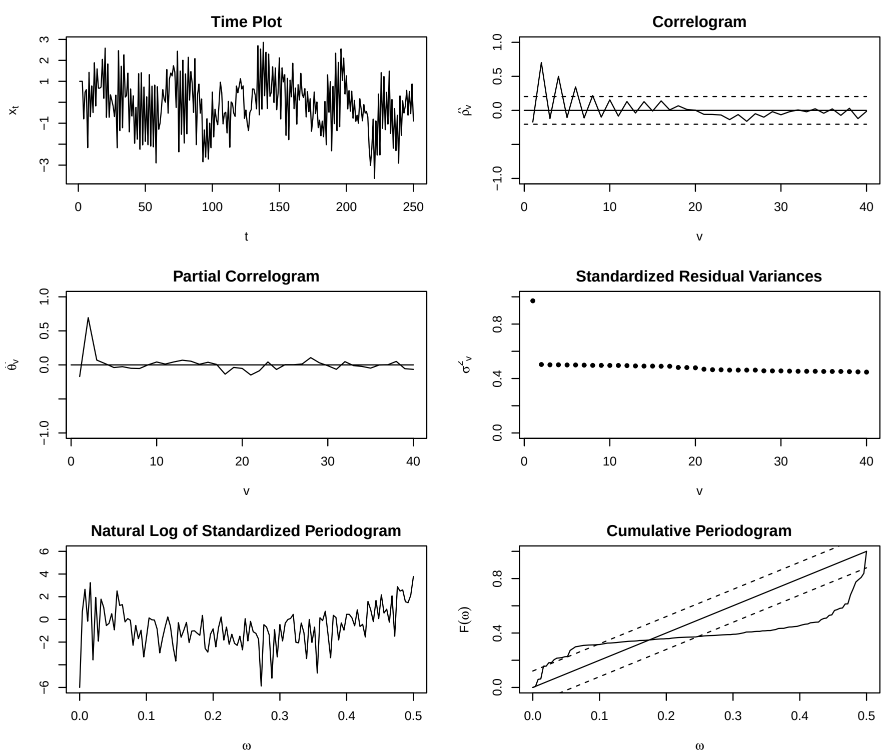




We reconstruct the series by:

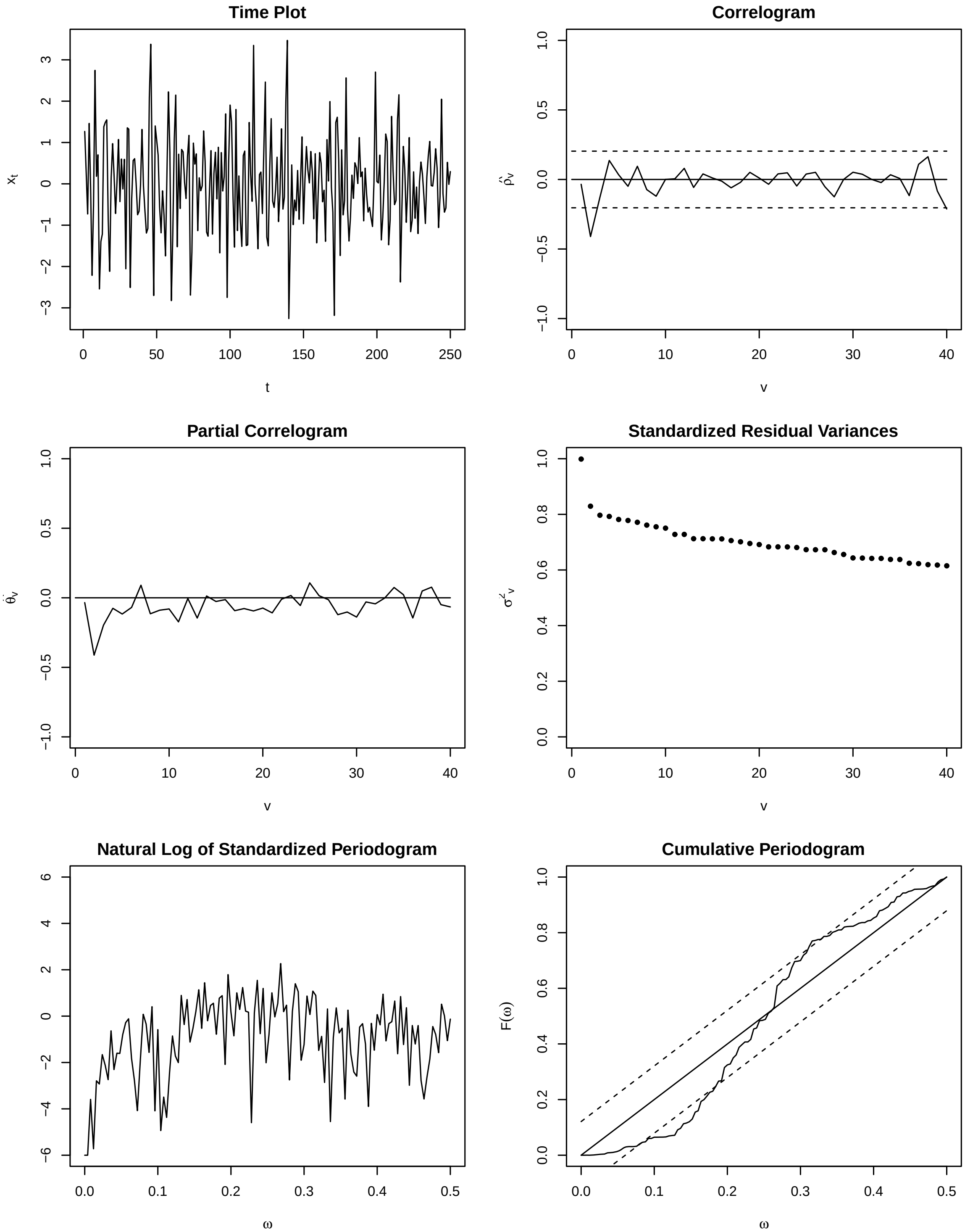
```
AR <- function(n=250, coef=c(0, 0, 0)){
	x = c(1,1,1)
	for (i in 4:n){
		x = c(x, coef[1]*x[i-1] + coef[2]*x[i-2] + coef[3]*x[i-3]+ rnorm(1))
	}
	return(x)
}

ar5 = AR(n=250, coef=c(0, 0.73, 0))
descplot(ar5, 40)
```



Since AR process is not deterministic we can get a new results each time we simulate it, thus generated plot doesn’t look exactly the same, yet it has all the same properties as the TS5.

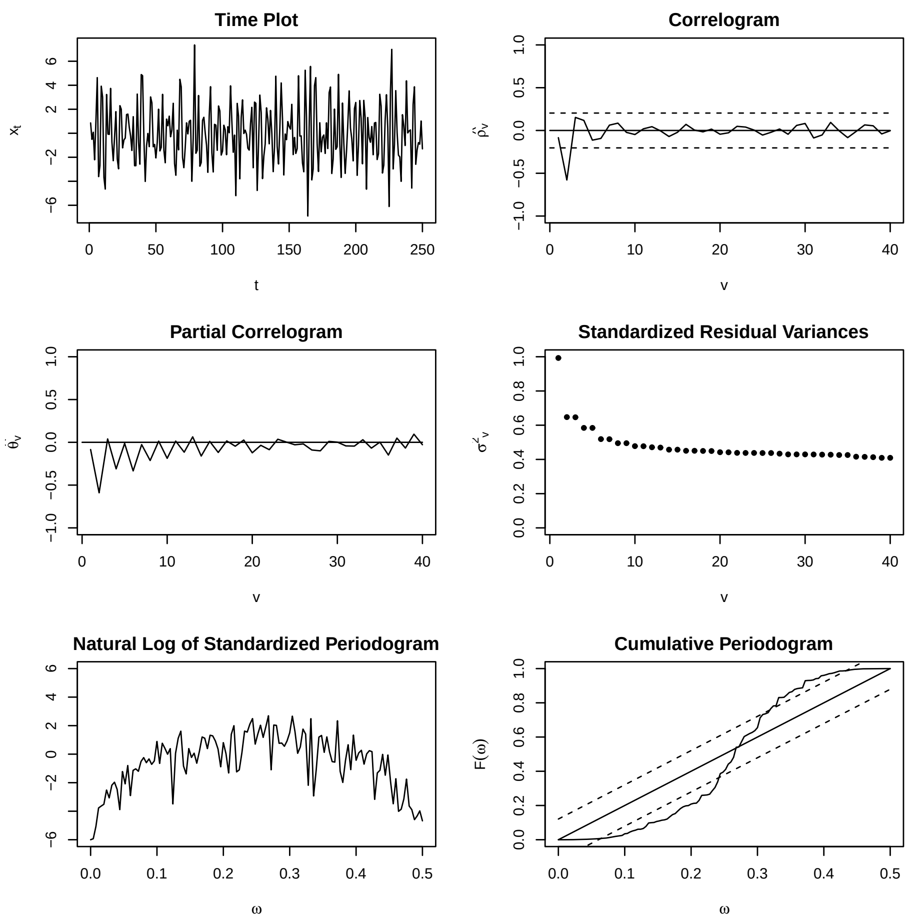
Series6

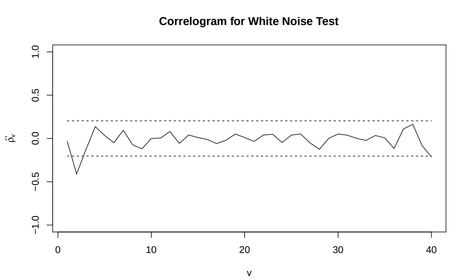


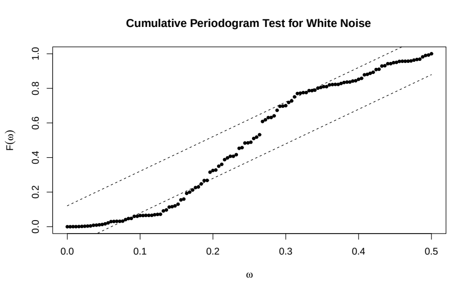
Looks almost like white noise. However middle frequencies made up most of the growth of cumulative periodogram. Residual variance contradicts the white noise since variance unexplained drops after lag=1. Overall, it looks like this series were generated by the **Moving Average** process ($x_t = 1.73*e_{t-1}+e_t$). Periodogram confirms this hypothesis with its smooth bump.

To reconstruct TS:

```
MA <- function(n=250, coef=c(0, 0, 0)){
      noise = rnorm(n + 3)
      x = coef[1] * (noise[4:(n+3)] - noise[3:(n+2)]) +
            coef[2] * (noise[4:(n+3)] - noise[2:(n+1)]) +
            coef[3] * (noise[4:(n+3)] - noise[1:n])
      return(x)
}
ma6 = MA(n=250, coef=c(0,1.73, 0))
descplot(ma6, 40)
```

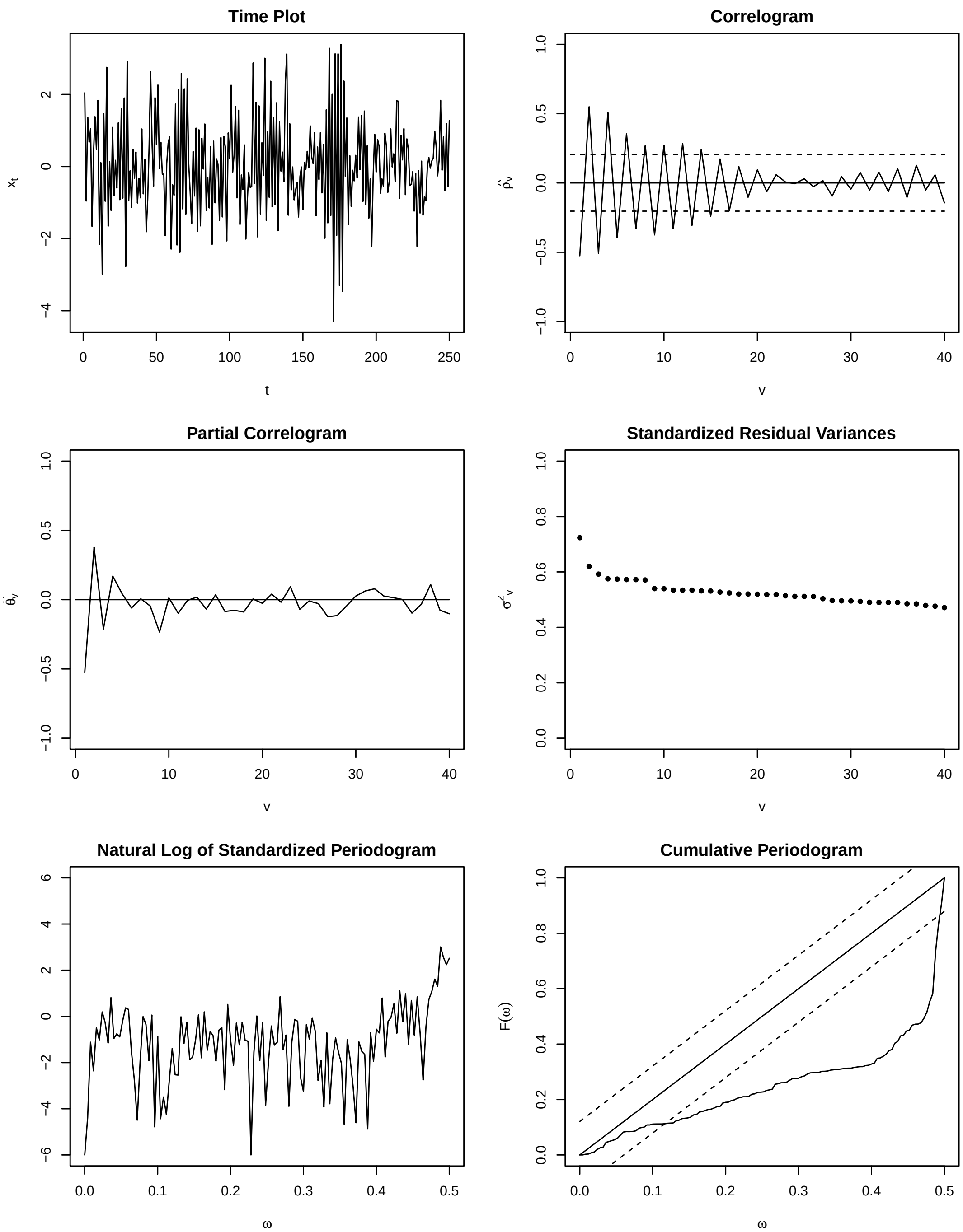






Again, looks almost like white noise, but not quite.

Series7



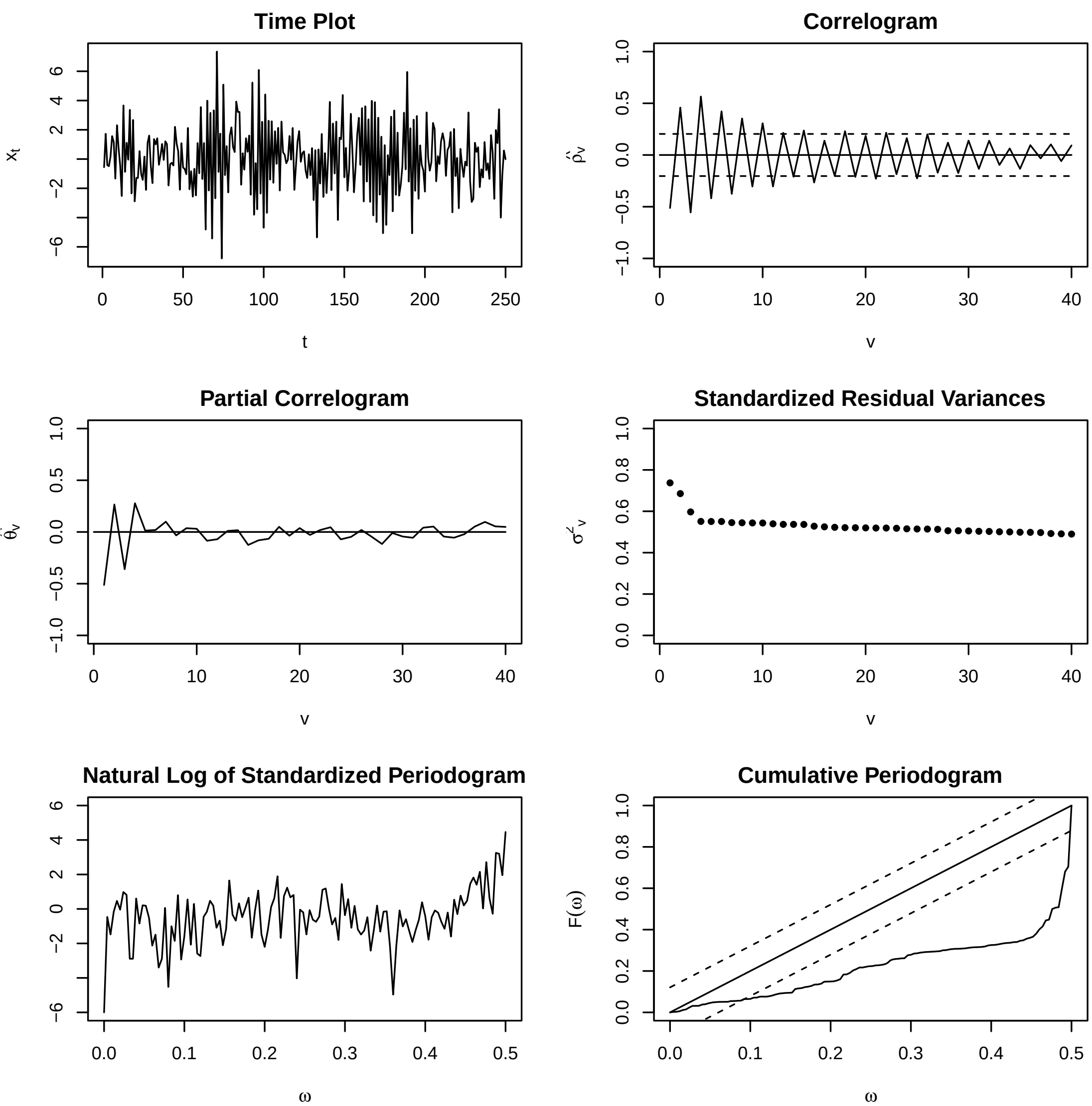
Doesn’t look like a white noise. Looks like something in between AR and MA. We can guess that it is **Autoregressive Moving Average.** Fluctuating correlogram supports this statement, as well as periodogram. The process might look like:

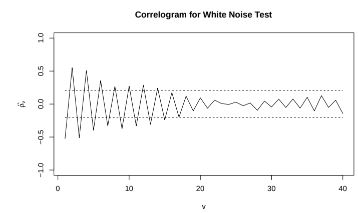
$\mathbf{x_t= -0.3x_{t-1} + 0.7x_{t-2} + 015x_{t-3} + 0.6*e_{t-2}+e_t}$

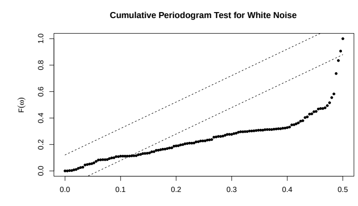
To reconstruct:

```
ARMA <- function(n=250, arc=c(0, 0, 0), mac=c(0, 0, 0)){
      x = c(1,1,1)
      noise = rnorm(n + 3)
      for (i in 4:(n+3)){
            x = c(x, arc[1]*x[i-1] + arc[2]*x[i-2] + arc[3]*x[i-3]+ noise[i])
      }
      noise = mac[1] * (noise[4:(n+3)] - noise[3:(n+2)]) +
      mac[2] * (noise[4:(n+3)] - noise[2:(n+1)]) +
      mac[3] * (noise[4:(n+3)] - noise[1:n])

      return(x[4:(n+3)] + noise)
}
arma7 = ARMA(arc=c(-0.3, 0.7, 0.15), mac=c(0, 0.6, 0))
descplot(arma7, 40)
```

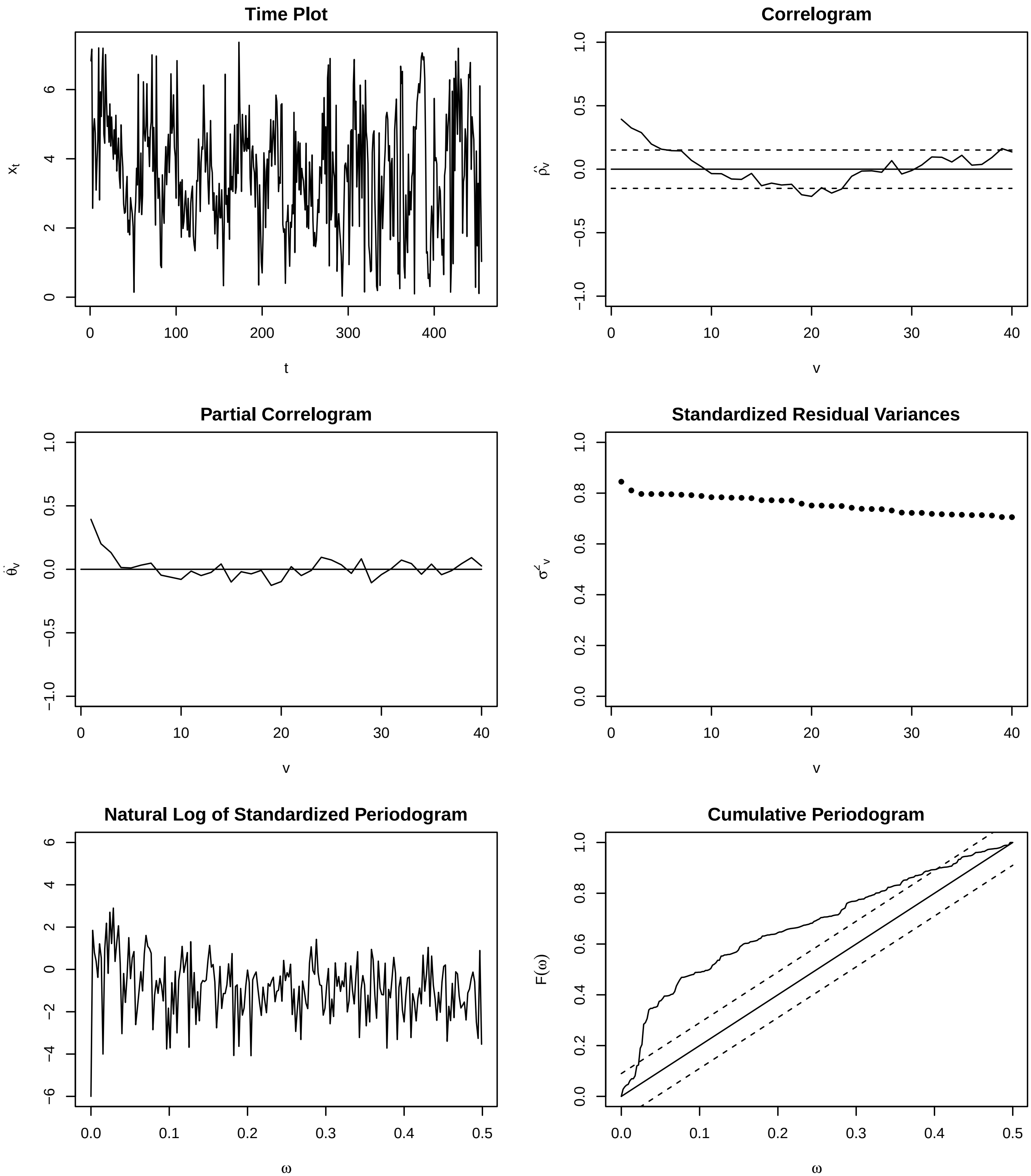






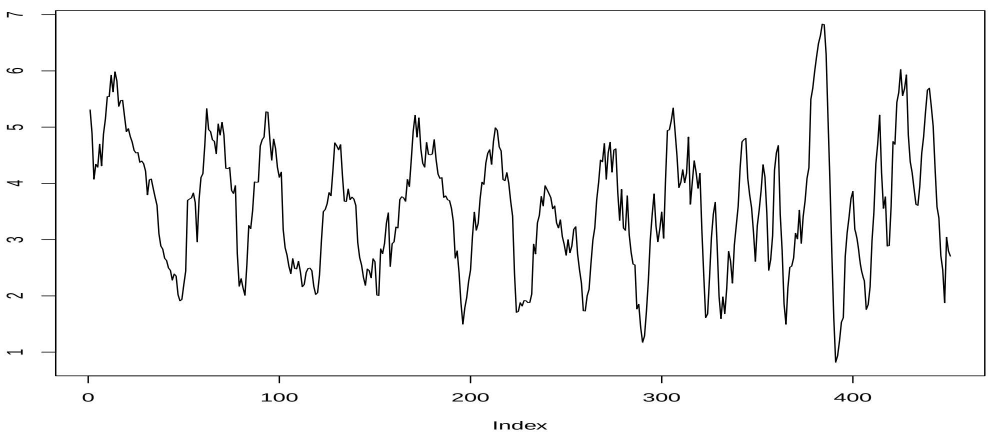
Doesn't look like a white noise. Reject H0.

Series8 (gkg – personal data)



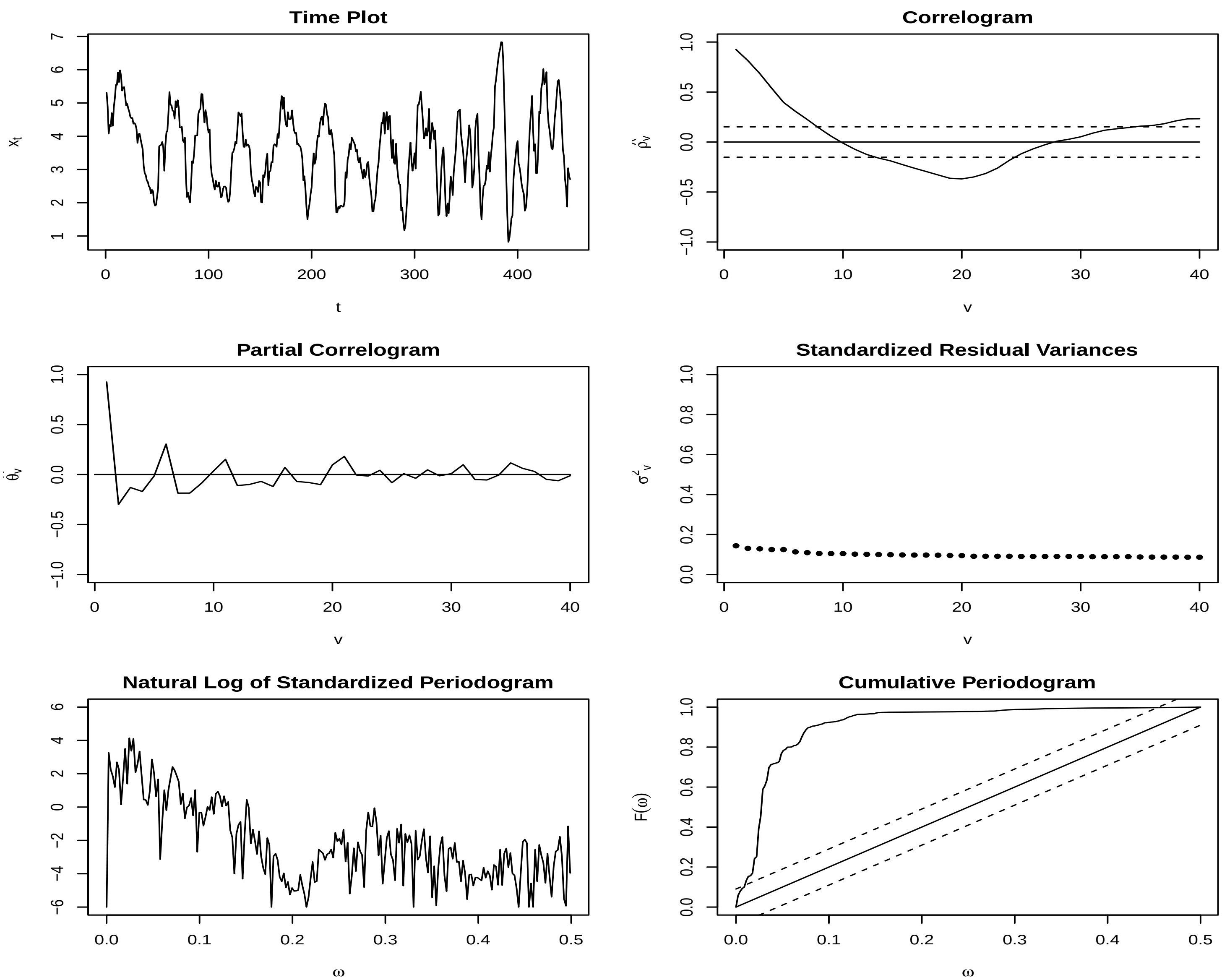
Not a white noise. However very hard to tell what's going on. Let's apply moving average smoother (with n=5):

```
plotl(rollmean(gkg, 5))
```



Now it is pretty obvious that we have some cycles, seemingly not linear trend. Let’s see descplot again:

```
descplot(rollmean(gkg, 5), 40)
```

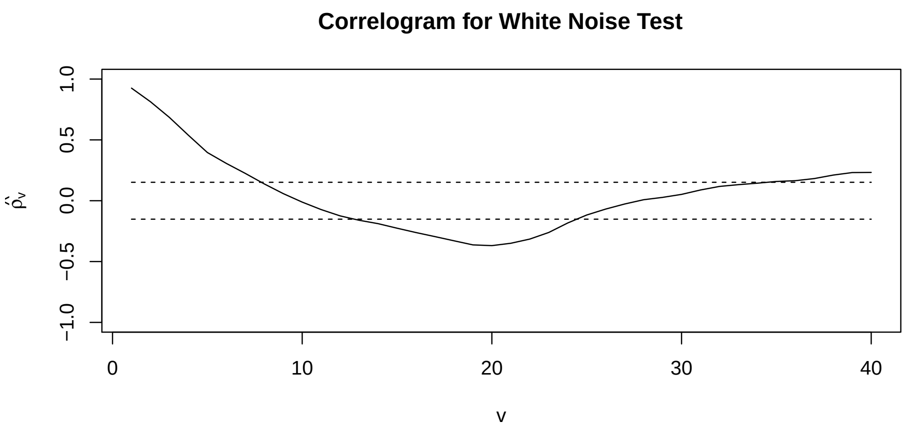


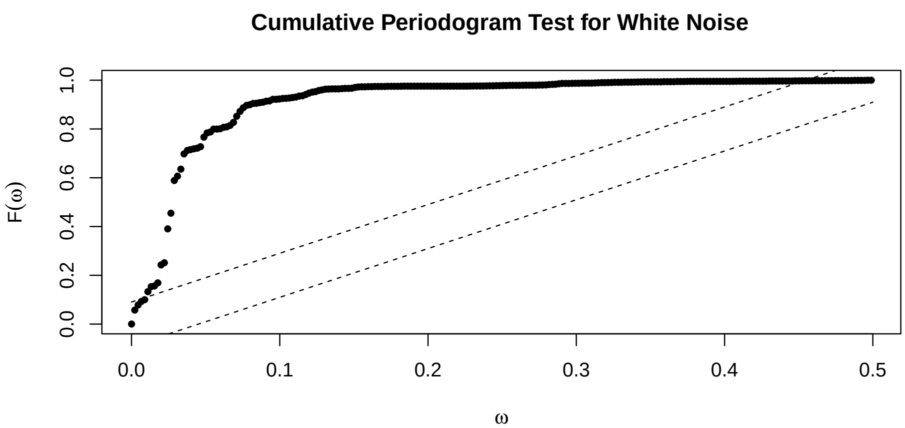
That looks like a mix of sinusoids. Like in the TS4 we didn’t catch the natural frequencies though.

Problematically, when trying to analyze the prevalent amplitudes and frequencies:

```
gkg = rollmean(gkg, 5)
gkg.freqs = round(Mod(fft(gkg))[1:(length(gkg)/2+1)] * 2 / length(gkg), 6)
gkg.freqs[gkg.freqs < 0.3] = 0
gkg.freqs[gkg.freqs > 0] # amps
[1] 0.386547   0.302347  0.387633  0.567712  0.507307  0.508044 0.337316
length(gkg) / (-1 + which(gkg.freqs > 0)) #freqs
[1] 447.00000  89.40000  49.66667  40.63636  37.25000  34.38462 27.93750
```

We don't get determined results despite we can see from the plot a fairly consistent pattern.





Clearly **not a white noise.**

TO BE CONTINUED with new models and tools