

Laboratory 2

Daniel Alonso

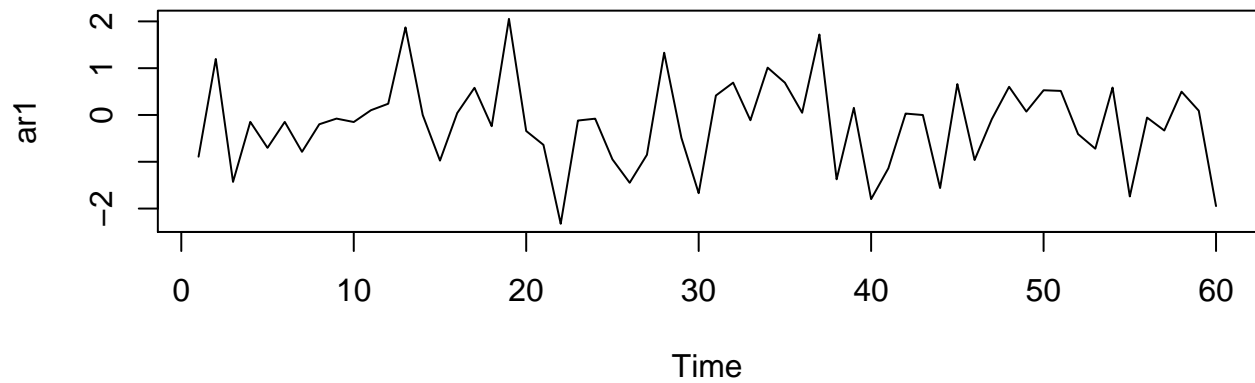
February 16, 2021

1. Use of command “arima.sim” to simulate ARIMA processes

$AR(1)$

We first simulate an $AR(1)$ process:

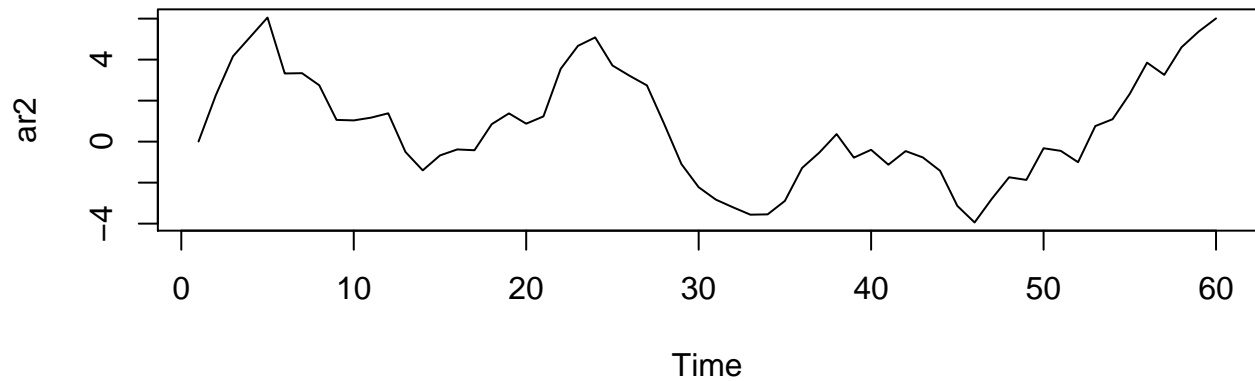
AR(1) simulation



$AR(2)$

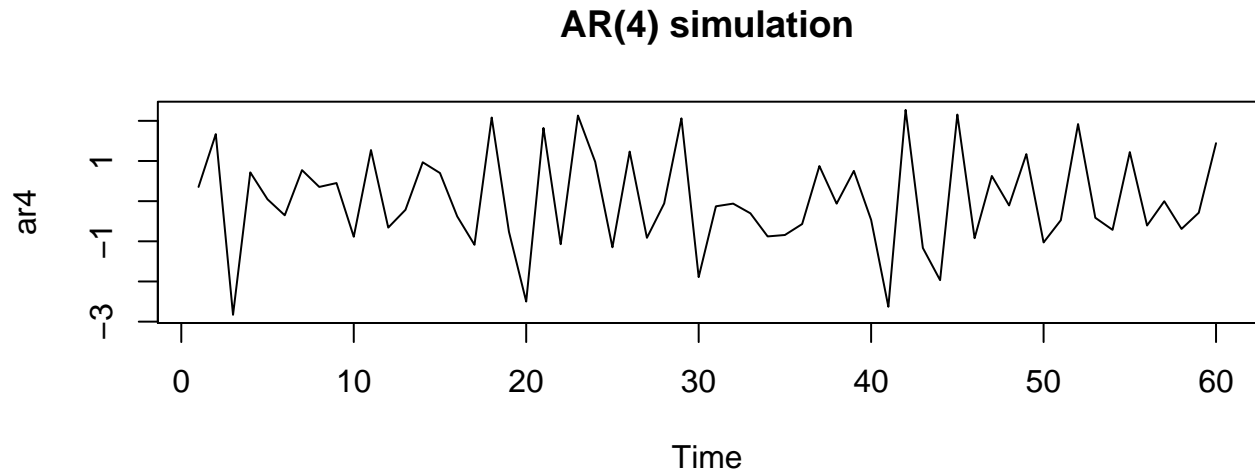
We then simulate an $AR(2)$ process:

AR(2) simulation



$AR(4)$

We then simulate an $AR(4)$ process:



2.simulates N $AR(p)$ processes ### With n observations,parameter ϕ standard deviation of $a = s$

```
arfun<-function(N,n,phi,s,c){
  M=matrix(ncol=N,nrow=n)
  for (i in 1:N){
    x=arima.sim(list(ar=phi),sd=s,n)
    M[,i]=x+c
  }
  #Computes mean, variance ACF and PACF
  variance=matrix(ncol=N,nrow=1)
  m=matrix(ncol=N,nrow=1)
  rho=matrix(ncol=N,nrow=25)
  pi=matrix(ncol=N,nrow=24)
  for(i in 1:N){
    variance[i]=var(M[,i])
    m[i]=mean(M[,i])
    r=acf(M[,i], lag.max=24,plot=FALSE)
    rho[,i]=r$acf
    pr=acf(M[,i],lag.max=24,type="partial",plot=FALSE)
    pi[,i]=pr$acf
  }
  #boxplots for the ACF and PACF of lags 1 to 4
  par(mfrow=c(2,2))
  boxplot(rho[2,],rho[3,],rho[4,],rho[5,], main="ACF coefficients for lags 1 to 4")
  boxplot(pi[1,],pi[2,],pi[3,],pi[4,], main="PACF coefficients for lags 1 to 4")
  plot(variance[1:N],type="l", main="Variance of the generated processes")
  plot(m[1:N],type="l", main="Mean of the generated processes")
}
```

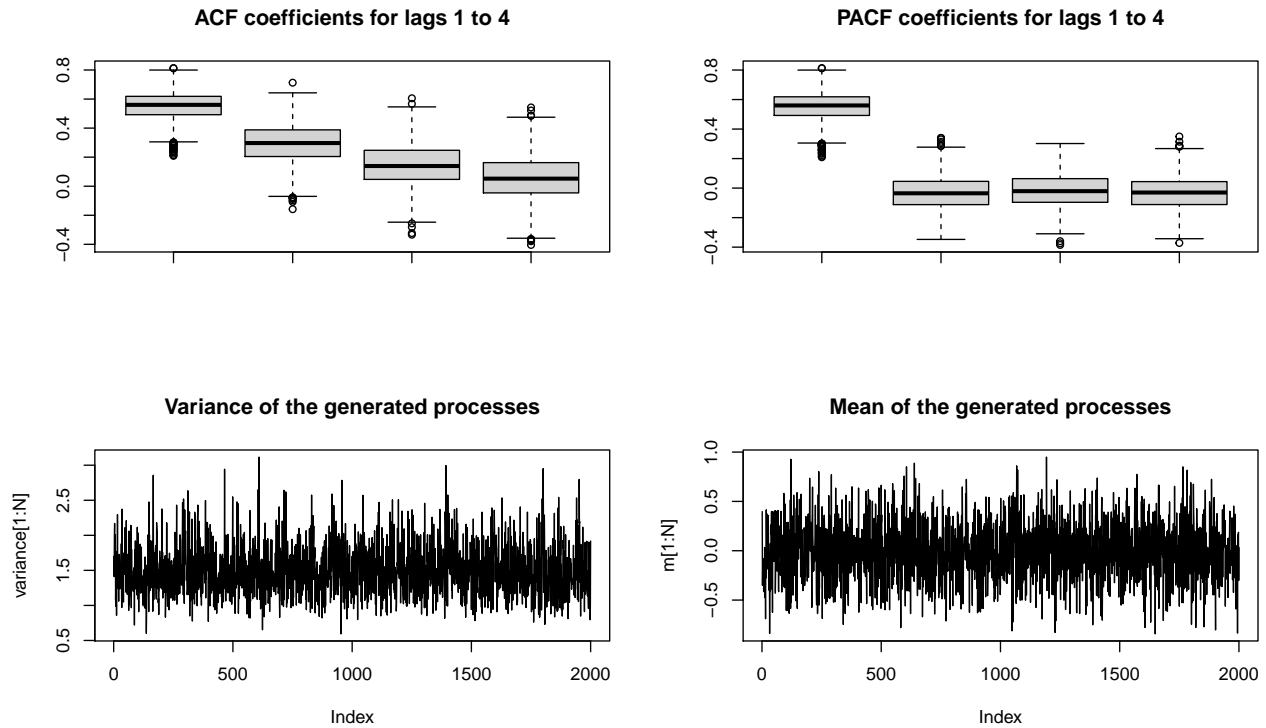
Simulating $AR(1)$ process with arfun

For all simulations we use 2000 iterations of 70 observations and $c = 0$.

Using a positive coefficient

With $\phi = 0.6$.

```
arfun(2000, 70, 0.6, 1, 0)
```



We can see that the first few ACF coefficients are still showing a ladder-like structure. The median of the first boxplot in both the ACF and PACF plots is of almost ~ 0.6 .

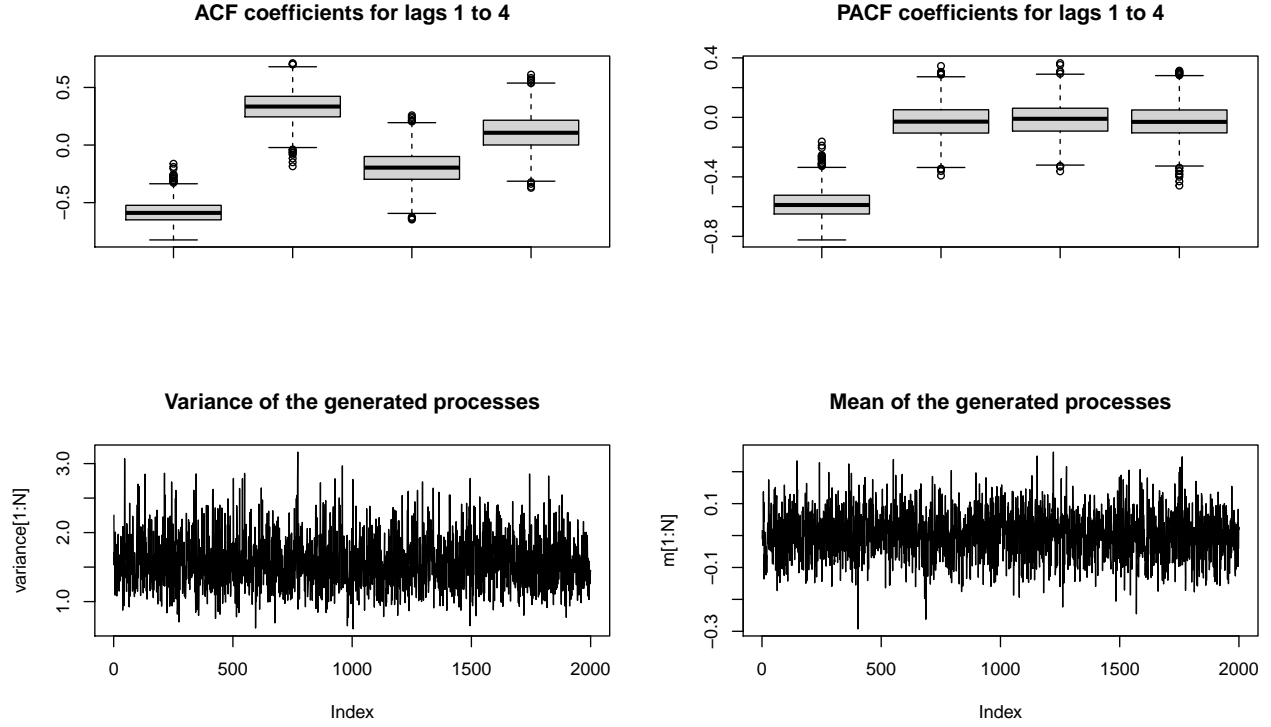
The variance is not huge, and from testing we know that it's less and less significantly variable as the number of samples and simulations increase simultaneously. Bias is also reduced significantly as these values increase. In fact, for a very large number of simulations, we notice no significant ACF or PACF coeffs.

The means oscillate around -0.3 and 0.3, while the variance goes to just under 1 up to a bit over 3.

Using a negative coefficient

With $\phi = -0.6$

```
arfun(2000, 70, -0.6, 1, 0)
```



The pattern for the ACF plot oscillates as the powers of a negative value change sign depending on whether the exponent is even or odd. For the PACF coefficients, only the first one is significant. For the first PACF coefficient with a negative ar , the value is pretty much ~ -0.6 .

The means oscillate around -0.2 and 0.2 with some outliers, while the variance is between 0.5 and 3.5 with most of the cases between 1 and 3.

Here, in contrast to a positive ϕ , the significance of our ACF coefficients is not exactly smoothed down, and ACF coefficients remain significant, while only the first PACF coefficient is significant.

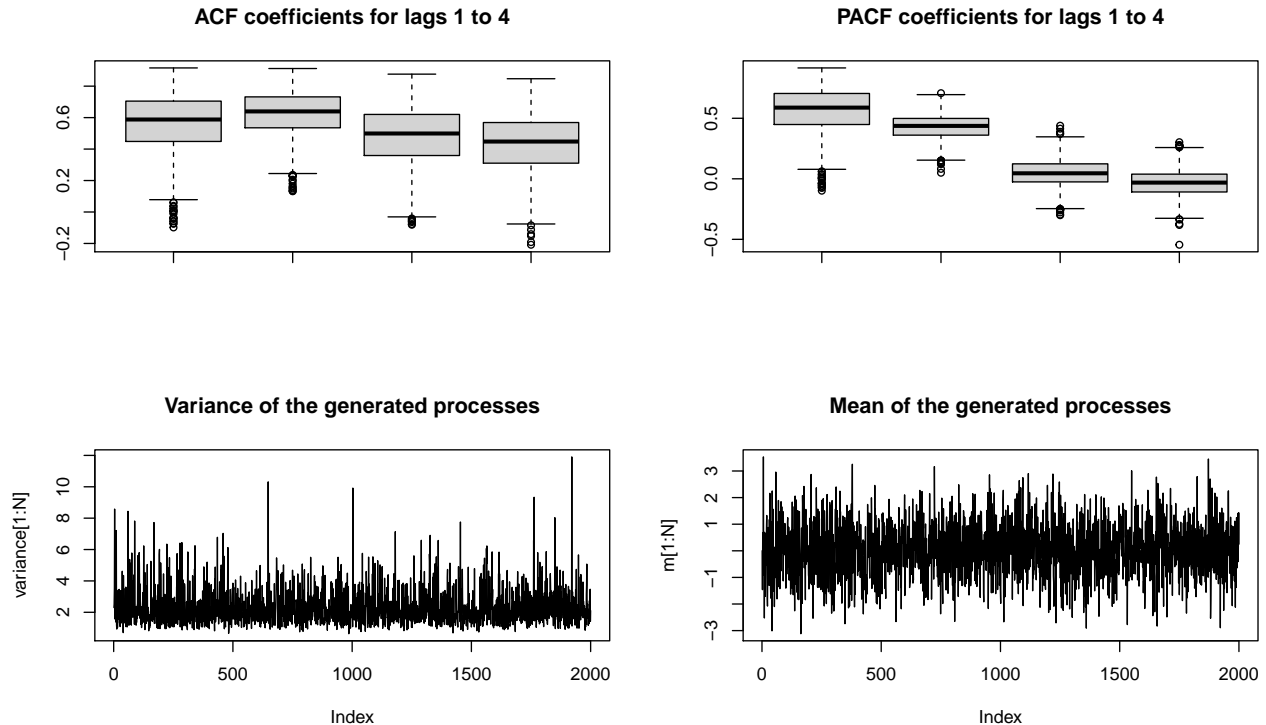
Simulating $AR(3)$ process with arfun

I have decided to include an $AR(3)$ process to comparatively see what happens to significance in ACF and PACF plots, along with variance and mean, when the signs of ϕ change, with multiple ϕ . All simulations are performed using 2000 iterations, 70 observations and $c = 0$.

Using 3 positive coefficients

With $\phi_1 = 0.3, \phi_2 = 0.5$ and $\phi_3 = 0.1$

```
arfun(2000, 70, c(0.3, 0.5, 0.1), 1, 0)
```



In this case we see ACF significance drops slowly vs the $AR(1)$ case. For PACF only maybe our first and second coefficients are significant.

The variance has much more extreme values and hovers (mostly) between 1 and 4, however, there's plenty of peaks, and all much more significant than those of the $AR(1)$ case.

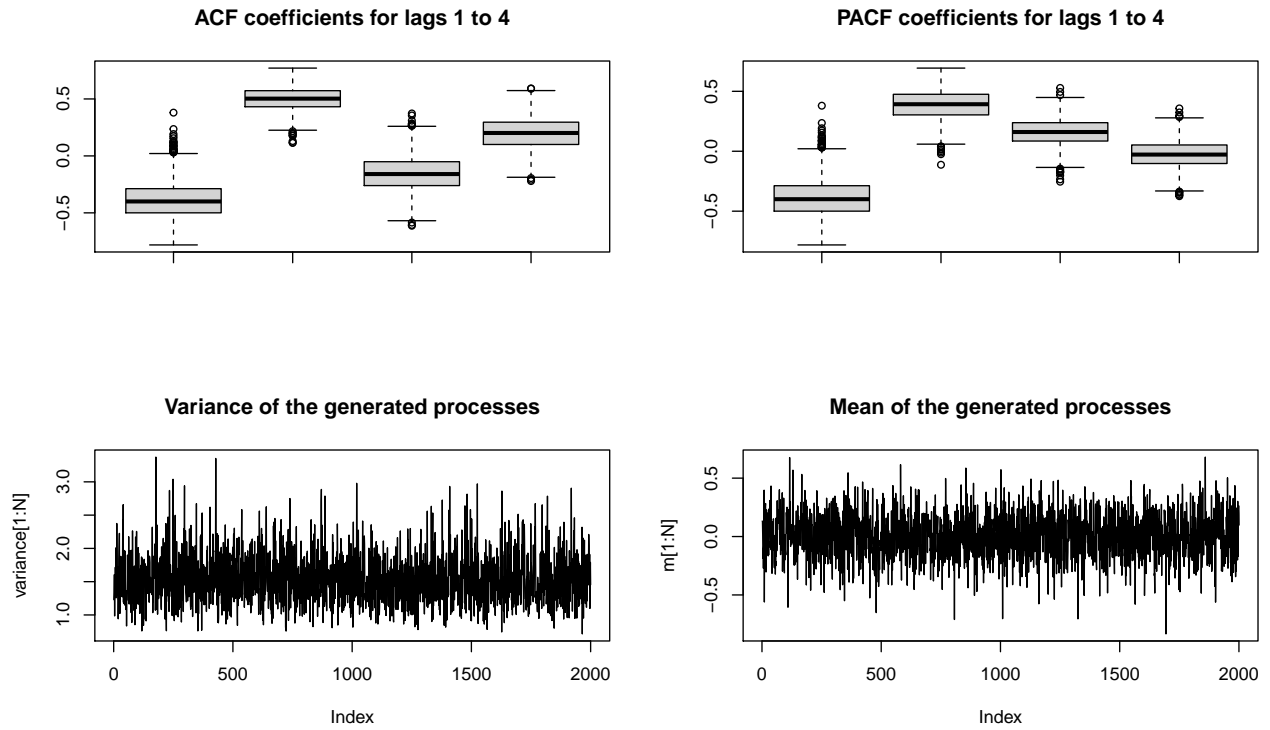
For the mean we notice a similar trend, however values tend to reach outside the interval $y = (1, -1)$ much more often.

For a significantly higher amount of iterations, we notice that we increase the chances for high variance events to occur.

Using 2 positive and 1 negative coefficients

With $\phi_1 = -0.3, \phi_2 = 0.5$ and $\phi_3 = 0.1$

```
arfun(2000, 70, c(-0.3, 0.5, 0.2), 1, 0)
```

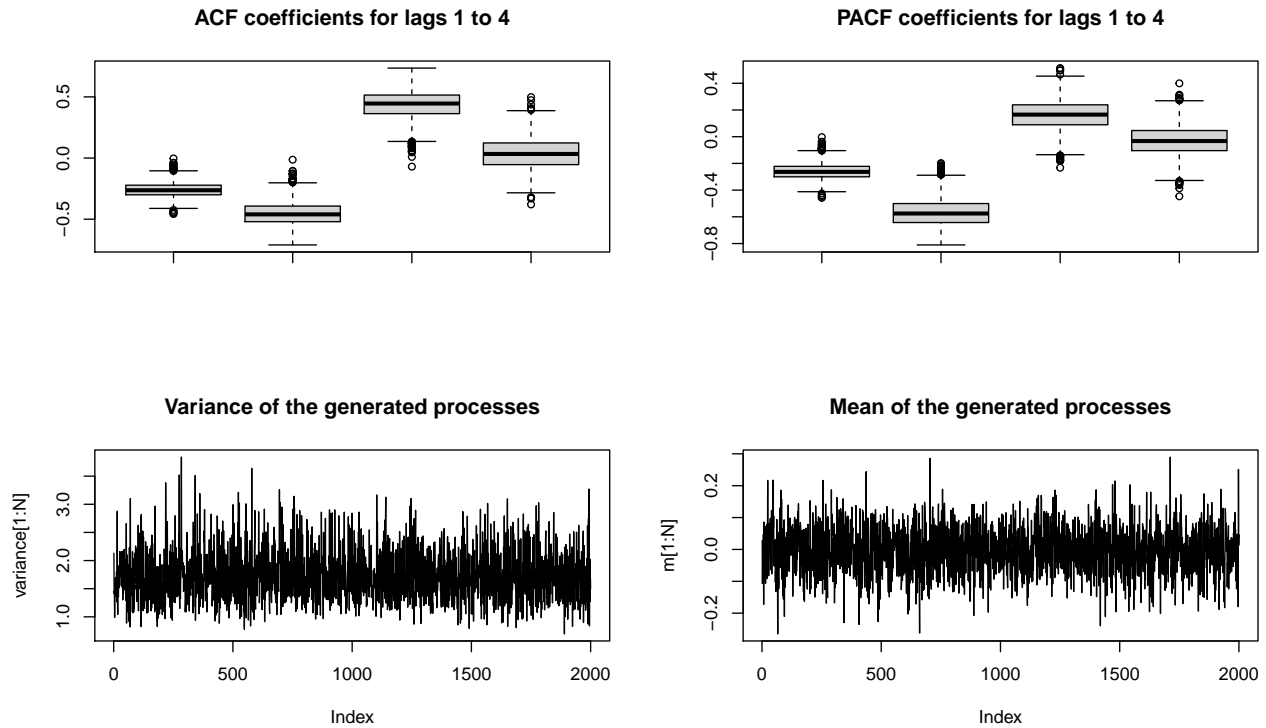


In the case where ϕ_1 is negative while the rest are positive, we notice the variance is significantly decreased, while our ACF and PACF coefficients remain significant for the first two lags. However they drop down quite quickly for the subsequent lags.

Using 1 positive and 2 negative coefficients

With $\phi_1 = -0.3, \phi_2 = -0.5$ and $\phi_3 = 0.1$

```
arfun(2000, 70, c(-0.3, -0.5, 0.2), 1, 0)
```



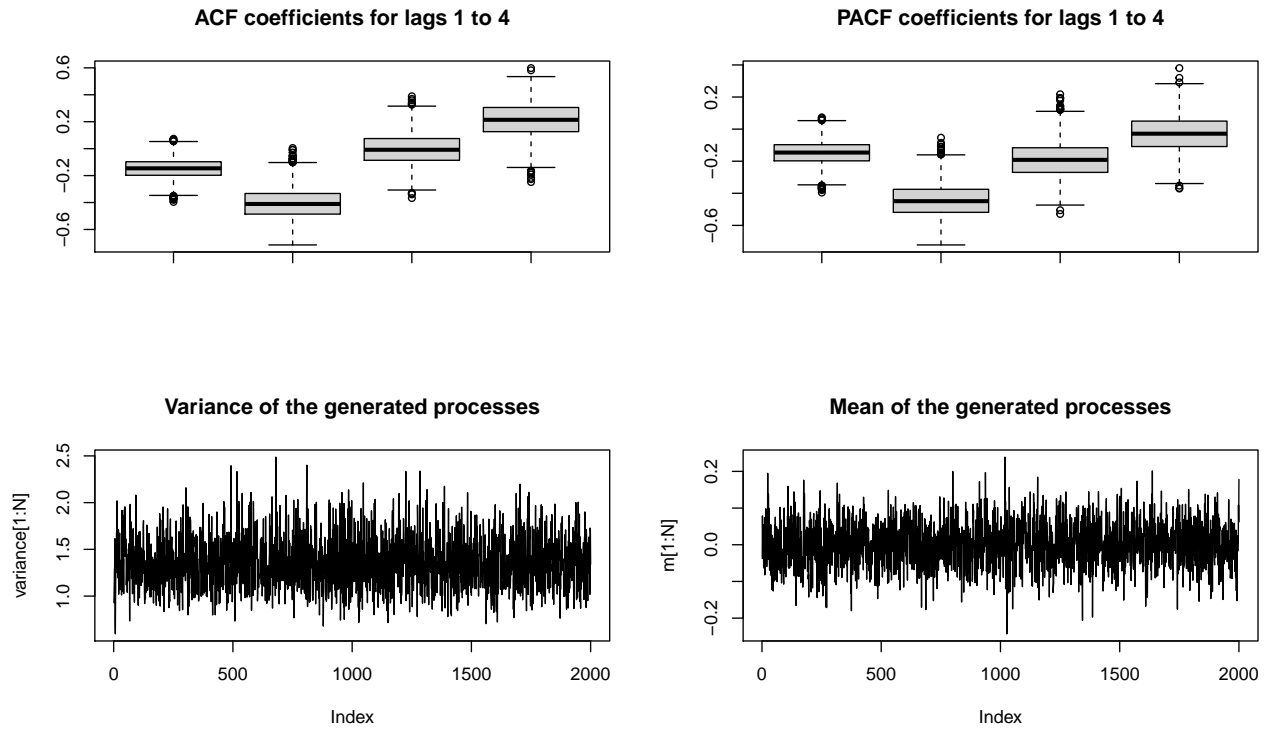
This option also keeps variance lower, overall, while maybe slightly higher than the previous option. The mean is significantly less variable overall.

The ACF and PACF coefficients seem to be boosted for the second lag vs the previous allocation of ϕ . Significance of such coefficients oscillates, as usual when using any negative ϕ , however it drops quite quickly for PACF, where the second lag is more significant than the first one.

Using 3 negative coefficients

With $\phi_1 = -0.3, \phi_2 = -0.5$ and $\phi_3 = -0.1$

```
arfun(2000, 70, c(-0.3, -0.5, -0.2), 1, 0)
```

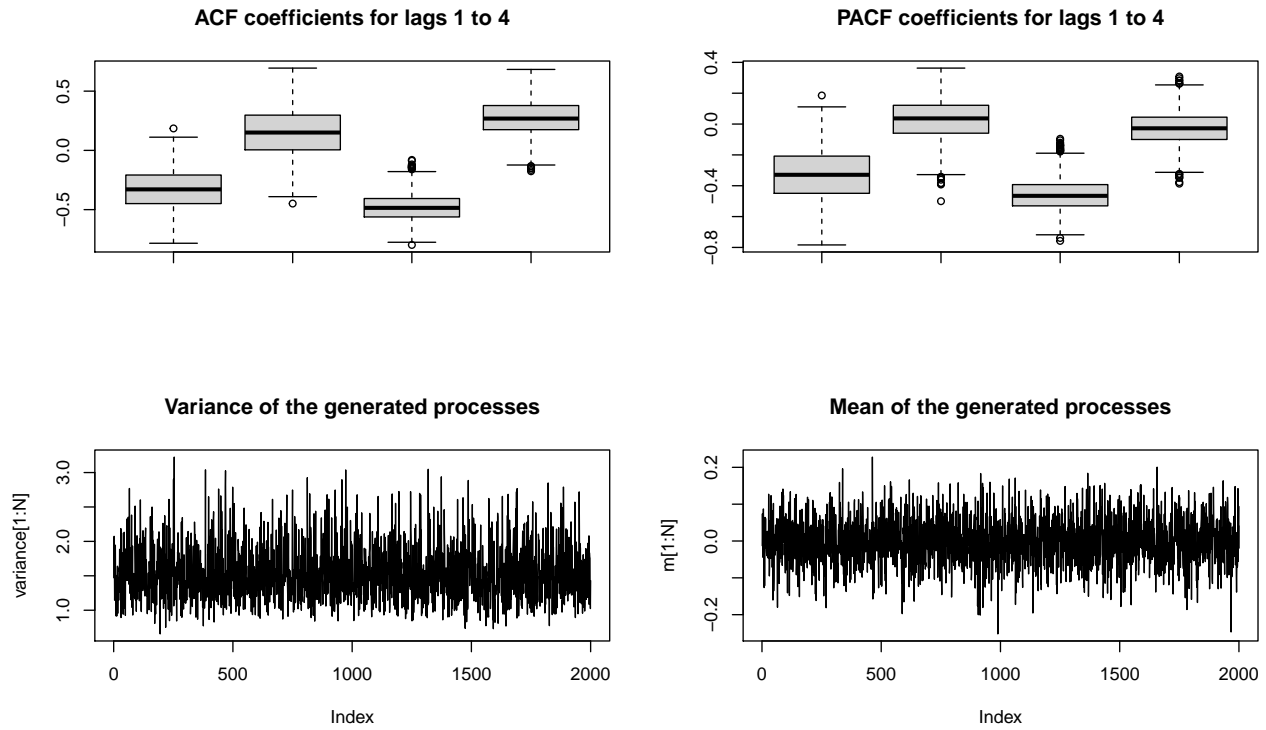


When using 3 negative coefficients we yield an even lower mean, and much less variant one as well. Our variance is overall also significantly reduced. Significance for ACF coefficients drops in general as well, making them all perhaps less significant. Similarly with PACF coefficients, with the exception of perhaps the 2nd lag.

Playing with values slightly

With $\phi_1 = -0.3$, $\phi_2 = -0.1$ and $\phi_3 = -0.5$

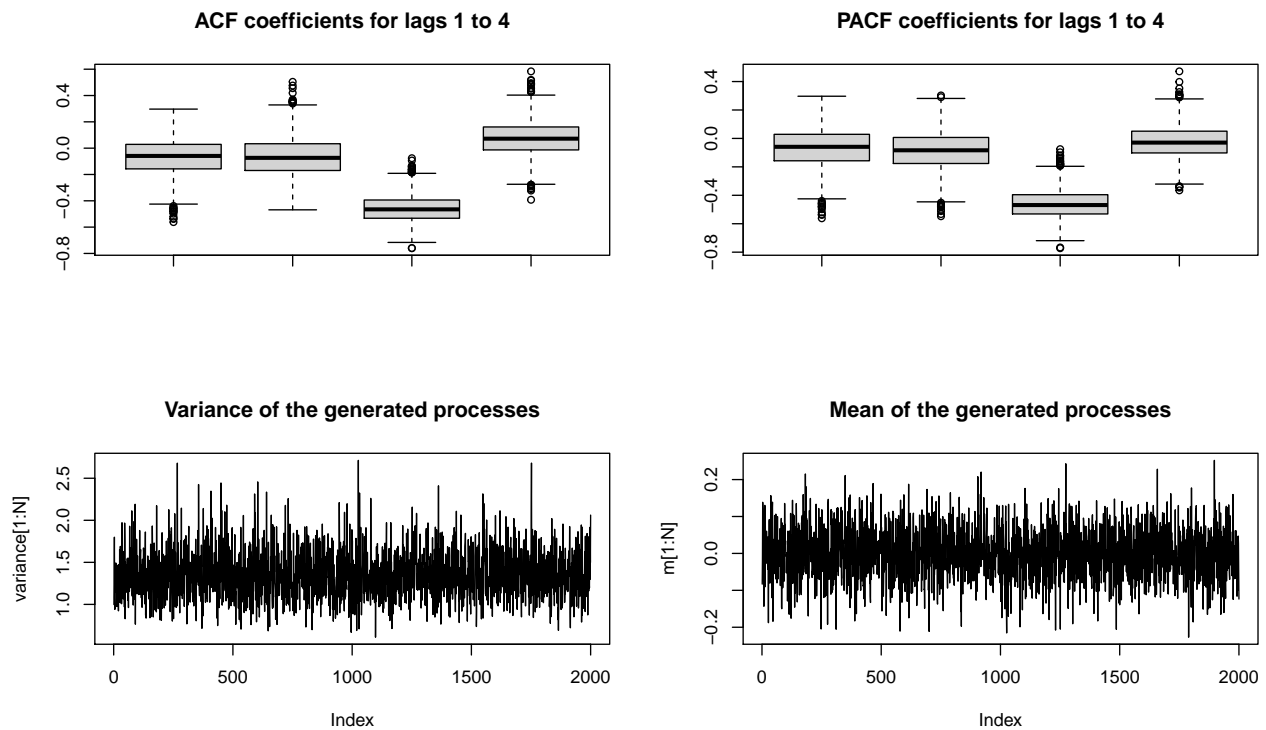
```
arfun(2000, 70, c(-0.3, -0.1, -0.5), 1, 0)
```

Changing which ϕ is larger/smaller also yields an interesting result, where the 1st and 3rd lags seem to show the most significance for both ACF and PACF plots.

With $\phi_1 = -0.3, \phi_2 = -0.1$ and $\phi_3 = -0.5$

```
arfun(2000, 70, c(-0.1, -0.1, -0.5), 1, 0)
```



Leveling down ϕ_1 and ϕ_2 to be the same value and also much higher than ϕ_3 basically removes the significance

of the first lag.