Danmarks
Tekniske
Universitet

Time Series Analysis
Assignment 2

Authors

Ghassen Lassoued - s196609

March 26, 2020

Contents

1 Question 2.1

Let the process X_t be given by:

$$X_t - 0.8X_{t-1} = \epsilon_t + 0.4\epsilon_{t-1} - 0.3\epsilon_{t-2} \tag{1}$$

which can also be written as

$$\phi(B)X_t = \theta(B)\epsilon_t \tag{2}$$

where ϵ_t is a white noise process with $\sigma = 0.4$ That gives : p= 1 and q= 2. It can be concluded then that X_t is an ARMA$(p = 1, q = 2)$ process.

1)

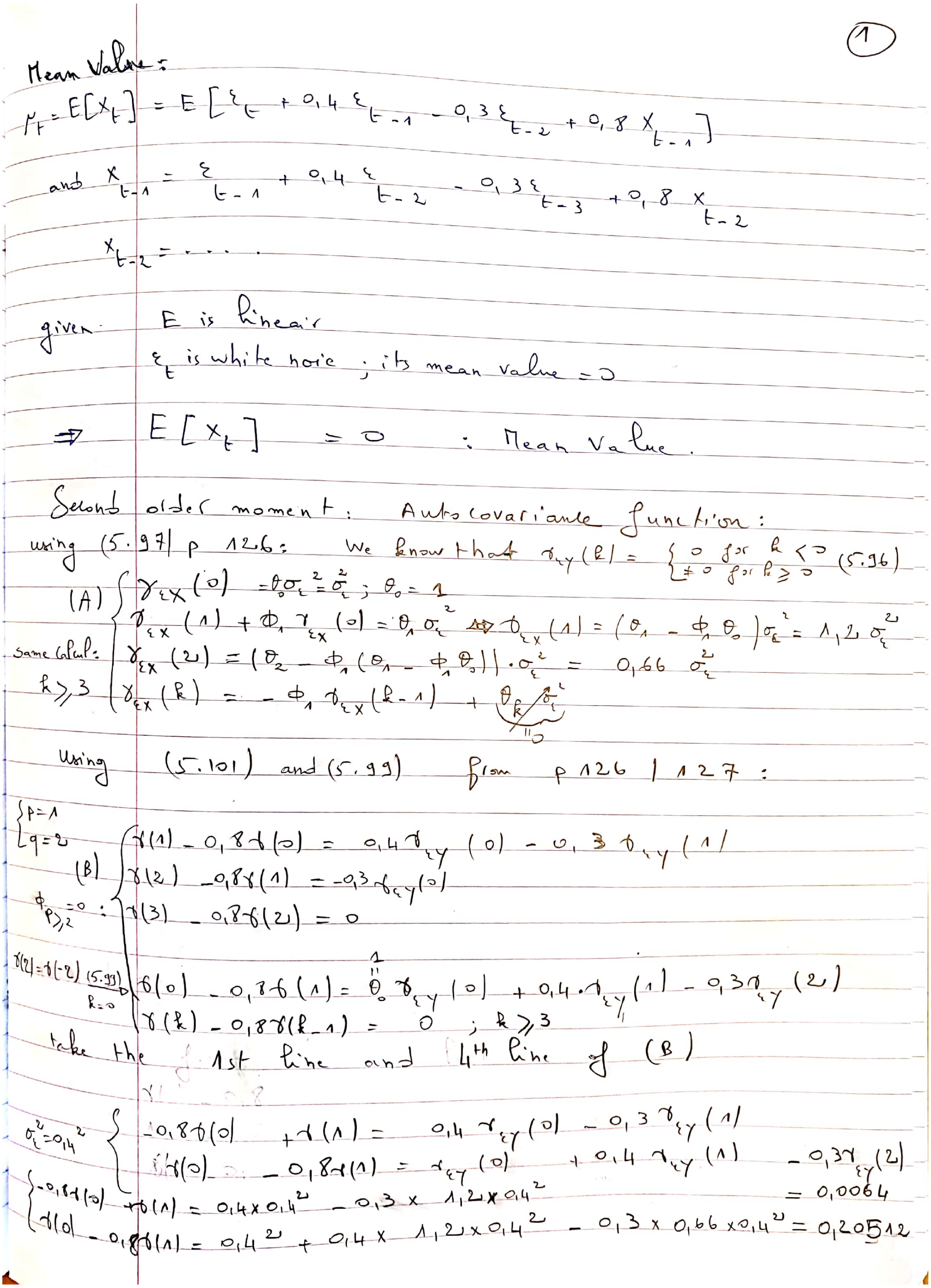
$$\phi(z^{-1}) = 1 - 0.8z^{-1} = 0 gives z = 0.8 < 1 \tag{3}$$

roots of $\phi(z^{-1}) = 0$ lie within the unit circle so the process is stationary

$$\theta(z^{-1}) = 1 + 0.4z^{-1} - 0.3z{-2} = 0 \tag{4}$$

After multiplying the equation with z^2 we get a second order polynomial which roots are [0.3830952 , 0.7830952]. The roots are calculated in R . roots of $\theta(z^{-1}) = 0$ lie within the unit circle so the process is invertible

2)



(1)

Mean Value:

$\mu_t = E[X_t] = E[\varepsilon_t + 0{,}4\,\varepsilon_{t-1} - 0{,}3\,\varepsilon_{t-2} + 0{,}8\,X_{t-1}]$

and $X_{t-1} = \varepsilon_{t-1} + 0{,}4\,\varepsilon_{t-2} - 0{,}3\,\varepsilon_{t-3} + 0{,}8\,X_{t-2}$

$X_{t-2} = \dots$

given: E is linear

ε_t is white noie ; its mean value $= 0$

$\Rightarrow \quad E[X_t] = 0$: Mean Value.

Second older moment: Autocovariance function:

using (5.97) p 126: We know that $\gamma_{\varepsilon Y}(k) = \begin{cases} 0 & \text{for } k < 0 \\ \neq 0 & \text{for } k \geq 0 \end{cases}$ (5.96)

$$(A)\begin{cases} \gamma_{\varepsilon X}(0) = \theta_0\sigma_\varepsilon^2 = \sigma_\varepsilon^2 \; ; \; \theta_0 = 1 \\ \gamma_{\varepsilon X}(1) + \phi_1\gamma_{\varepsilon X}(0) = \theta_1\sigma_\varepsilon^2 \Leftrightarrow \gamma_{\varepsilon X}(1) = (\theta_1 - \phi_1\theta_0)\sigma_\varepsilon^2 = 1{,}2\,\sigma_\varepsilon^2 \\ \text{same calcul:} \quad \gamma_{\varepsilon X}(2) = (\theta_2 - \phi_1(\theta_1 - \phi_1\theta_0))\cdot\sigma_\varepsilon^2 = 0{,}66\,\sigma_\varepsilon^2 \\ k \geq 3 \quad \gamma_{\varepsilon X}(k) = -\phi_1\gamma_{\varepsilon X}(k-1) + \underbrace{\cancel{\theta_k\sigma_\varepsilon^2}}_{=0} \end{cases}$$

Using (5.101) and (5.99) from p 126 / 127:

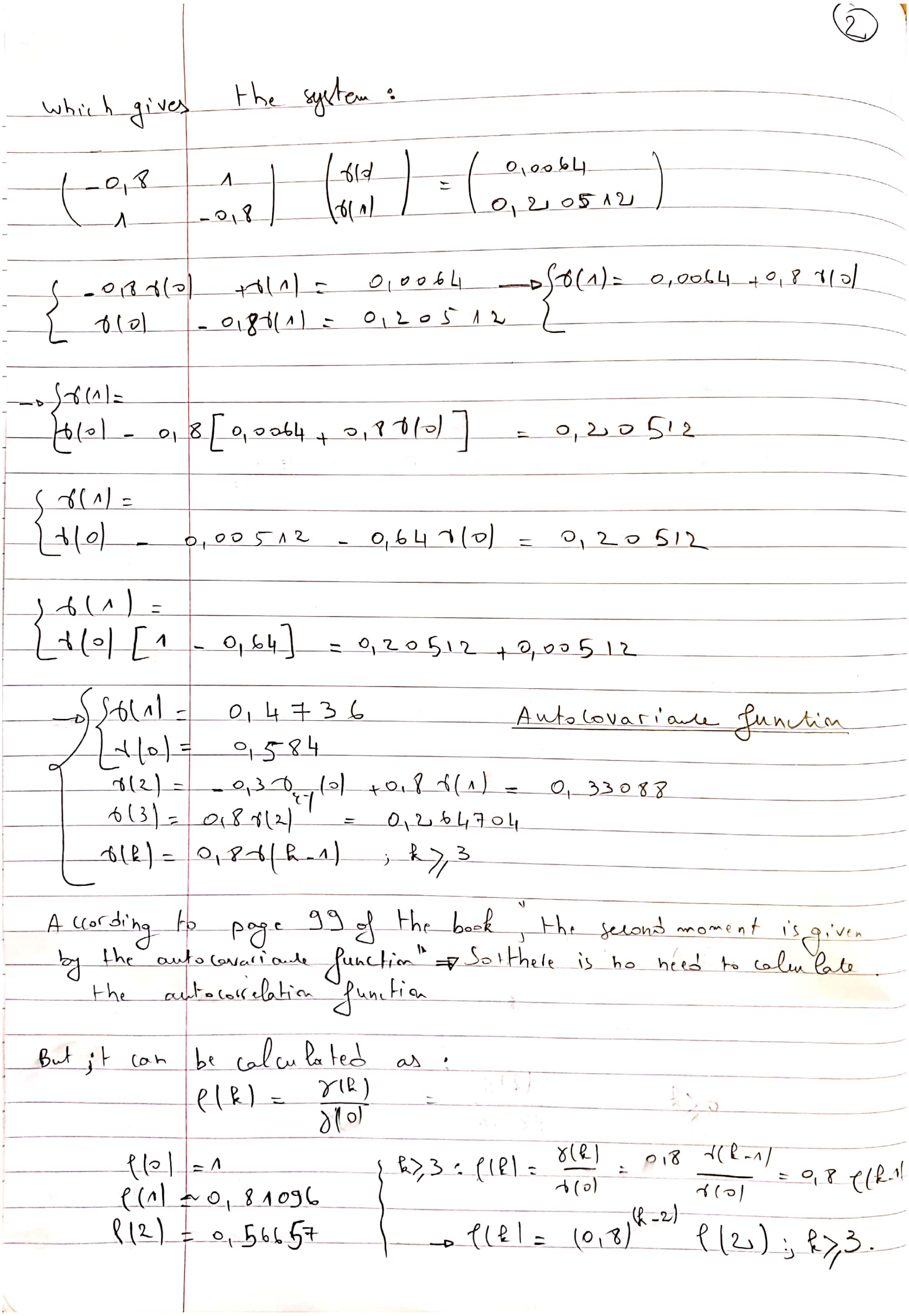
$\begin{cases} p = 1 \\ q = 2 \end{cases}$

$$(B)\begin{cases} \gamma(1) - 0{,}8\gamma(0) = 0{,}4\gamma_{\varepsilon Y}(0) - 0{,}3\gamma_{\varepsilon Y}(1) \\ \gamma(2) - 0{,}8\gamma(1) = -0{,}3\gamma_{\varepsilon Y}(0) \\ \phi_{p\geq 2} = 0: \quad \gamma(3) - 0{,}8\gamma(2) = 0 \\ \gamma(2) = \gamma(-2) \xrightarrow[k=0]{(5.99)} \gamma(0) - 0{,}8\gamma(1) = \overset{1}{\overset{\shortparallel}{\theta_0}}\gamma_{\varepsilon Y}(0) + 0{,}4\cdot\gamma_{\varepsilon Y}(1) - 0{,}3\gamma_{\varepsilon Y}(2) \\ \gamma(k) - 0{,}8\gamma(k-1) = 0 \; ; \; k \geq 3 \end{cases}$$

take the 1st line and 4th line of (B)

$$\sigma_\varepsilon^2 = 0{,}4^2 \begin{cases} -0{,}8\gamma(0) + \gamma(1) = 0{,}4\gamma_{\varepsilon Y}(0) - 0{,}3\gamma_{\varepsilon Y}(1) \\ \gamma(0) - 0{,}8\gamma(1) = \gamma_{\varepsilon Y}(0) + 0{,}4\gamma_{\varepsilon Y}(1) - 0{,}3\gamma_{\varepsilon Y}(2) \end{cases}$$

$$\begin{cases} -0{,}8\gamma(0) + \gamma(1) = 0{,}4\times 0{,}4^2 - 0{,}3\times 1{,}2\times 0{,}4^2 = 0{,}0064 \\ \gamma(0) - 0{,}8\gamma(1) = 0{,}4^2 + 0{,}4\times 1{,}2\times 0{,}4^2 - 0{,}3\times 0{,}66\times 0{,}4^2 = 0{,}20512 \end{cases}$$



(2)

which gives the system :

$$\begin{pmatrix} -0{,}8 & 1 \\ 1 & -0{,}8 \end{pmatrix} \begin{pmatrix} \gamma(0) \\ \gamma(1) \end{pmatrix} = \begin{pmatrix} 0{,}0064 \\ 0{,}205\,12 \end{pmatrix}$$

$$\begin{cases} -0{,}8\,\gamma(0) + \gamma(1) = 0{,}0064 \\ \gamma(0) - 0{,}8\,\gamma(1) = 0{,}20512 \end{cases} \longrightarrow \begin{cases} \gamma(1) = 0{,}0064 + 0{,}8\,\gamma(0) \\ \end{cases}$$

$$\longrightarrow \begin{cases} \gamma(1) = \\ \gamma(0) - 0{,}8\left[0{,}0064 + 0{,}8\,\gamma(0)\right] = 0{,}20512 \end{cases}$$

$$\begin{cases} \gamma(1) = \\ \gamma(0) - 0{,}00512 - 0{,}64\,\gamma(0) = 0{,}20512 \end{cases}$$

$$\begin{cases} \gamma(1) = \\ \gamma(0)\left[1 - 0{,}64\right] = 0{,}20512 + 0{,}00512 \end{cases}$$

$$\longrightarrow \begin{cases} \begin{cases} \gamma(1) = 0{,}4736 \\ \gamma(0) = 0{,}584 \end{cases} \\ \gamma(2) = -0{,}3\,\gamma_{\varepsilon}(0) + 0{,}8\,\gamma(1) = 0{,}33088 \\ \gamma(3) = 0{,}8\,\gamma(2) = 0{,}264704 \\ \gamma(k) = 0{,}8\,\gamma(k-1) \quad ; \ k \geqslant 3 \end{cases}$$

Autocovariance function

According to page 99 of the book, "the second moment is given by the autocovariance function" ⇒ So there is no need to calculate the autocorrelation function.

But, it can be calculated as :

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}$$

$$\rho(0) = 1$$
$$\rho(1) \approx 0{,}81096$$
$$\rho(2) = 0{,}56657$$

$$\begin{cases} k \geqslant 3 : \rho(k) = \dfrac{\gamma(k)}{\gamma(0)} = 0{,}8\,\dfrac{\gamma(k-1)}{\gamma(0)} = 0{,}8\,\rho(k-1) \\ \longrightarrow \rho(k) = (0{,}8)^{(k-2)}\,\rho(2) \ ; \ k \geqslant 3. \end{cases}$$

3)

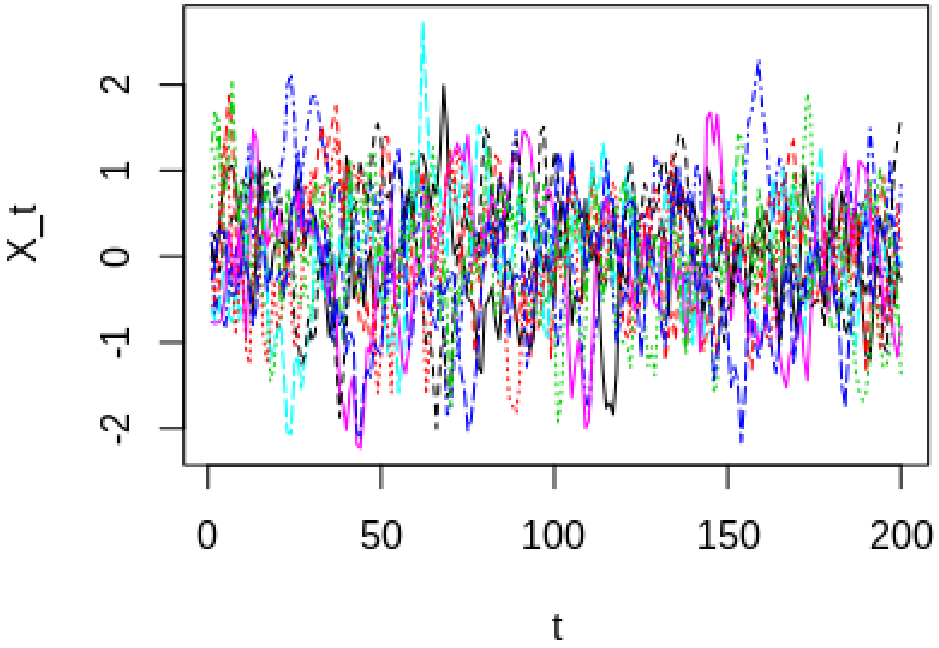


Figure 1: 10 realizations of the process

The plot of the 10 realizations reveales that they behave in a similar way. They all have ups and downs i.e there is no long term trend. That enhances the fact that they are stationary.

4)

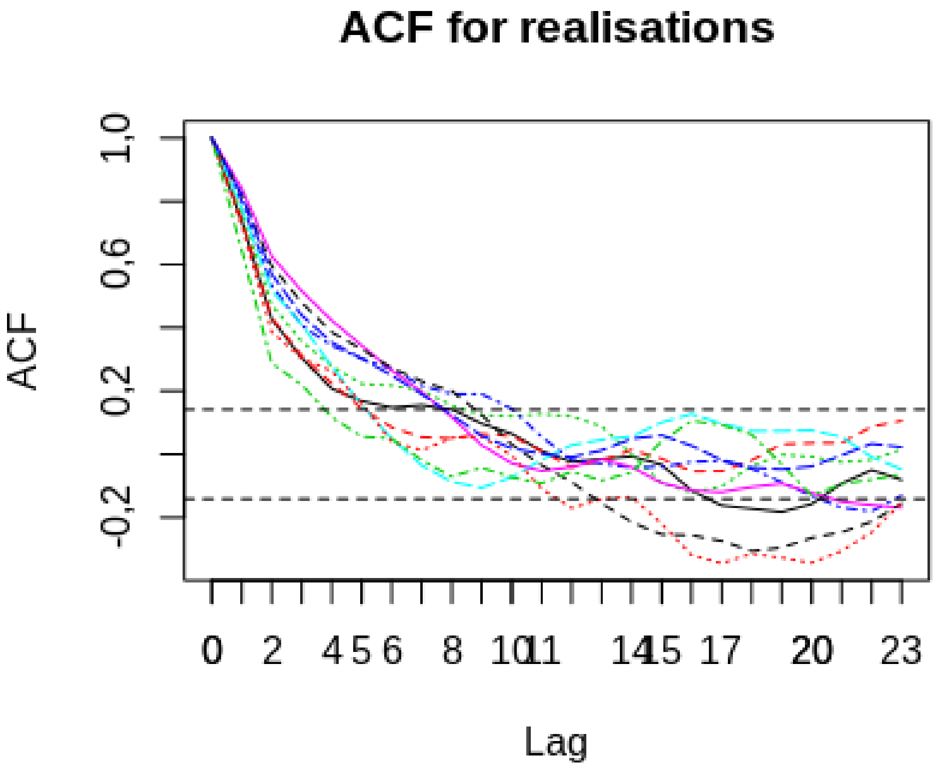


Figure 2: ACF of 10 realizations

ACF stands for autocorrelation function. In the plot above, a 95% confidence interval is given by $\pm 2/\sqrt{N} = 0.14142$ It is represented by the dashed line. The values of the autocorrelation function is not significant. It can be seen that after lag(q-p) =lag(1), the autocorrolation functions are damped exponentially according to the table 6.1 p155. That is the behaviour expected from an ARMA model and also the AR part in ARMA model.It is hard to see the sine behaviour before lag7. Regarding the MA part, it takes about 10 lags for all the realizations autocorrelation functions to be insignificant and to be equal to 0.
Remark:
-The root of the MA part is 0.7830952 which is close to 1. This root has a relation with the process being invertible. The question is does it explain the time 10 lags ?
-The autocorrelation functions of the 10 realizations are behaving in the same way agreeing with the process being stationary.
5)

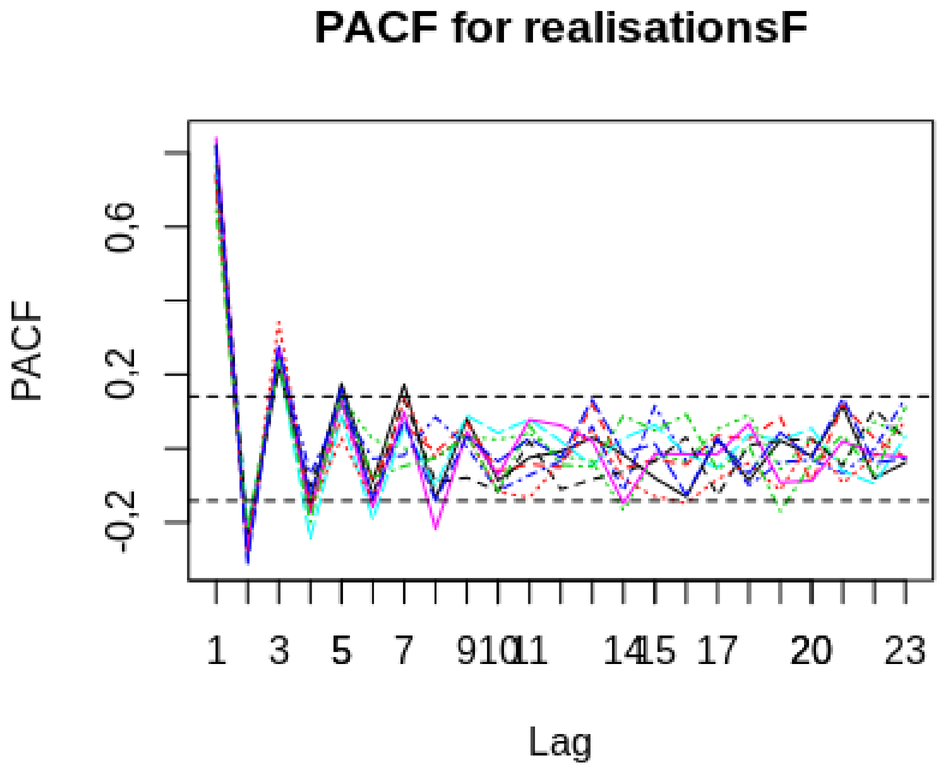


Figure 3: PACF of 10 realizations

PACF stands for partial autocorrelation function. The confidence interval is built in the same manner as before. The plot above shows that the partial autocorrelation functions of the 10 realizations are changing sign as expected and that they are decreasing exponentially after lag1. According to table 6.1 p 155 , this behaviour is obtained when p-q=1 corresponding to ARMA(2,1) process which is not the process in this case. That makes it hard to say which process is it from seeing this plot.
The partial autocorrelation functions of the 10 realizations are behaving in the same way agreeing with the process being stationary
6)

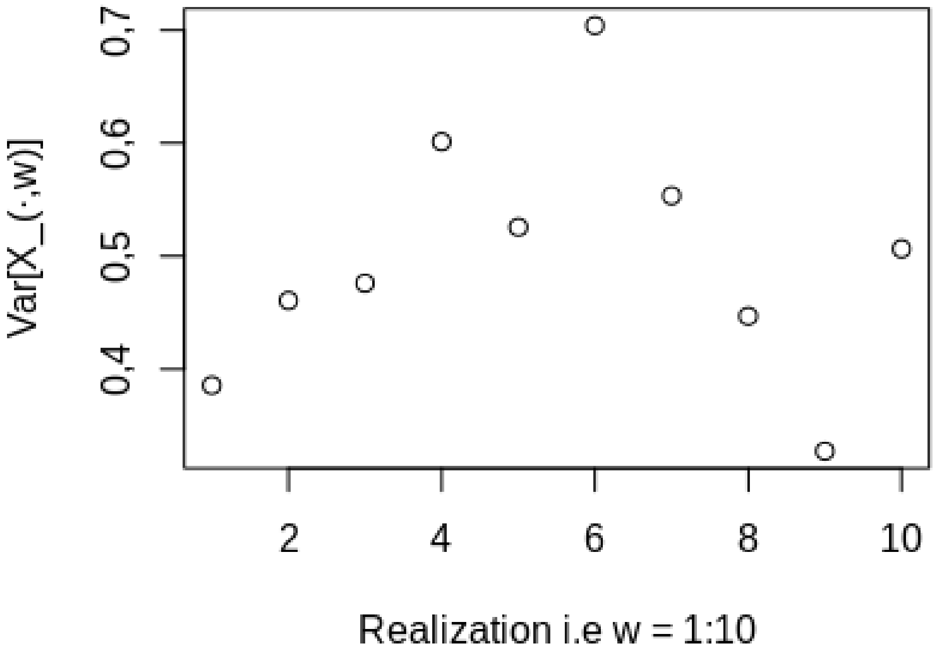


Figure 4: Variance of the process in 10 realizations

The variances of the process in the 10 realizations are :

[0,3856283 0,4606983 0,4759824 0,6012659 0,5255765 0,7040809 0,5533147 0,4468784 0,3272994 0,5064150]

The values of the 10 realizations are close. Analitically, the variance is $\gamma(0) = 0.584$ which is close to the mean of the 10 realizations 0,498714 . This agrees with the idea of ergodicity.

7) R gives estimated ACF and PACF . They are called sample autocorrelation function and sample partial autocorrelation function. The idea here is to compare them with the theoretical plot.

As said in 6) above, the numerical variances calculated are close to the theoretical variance. Regarding the ACF, the plot below includes the analytical ACF calculated in the last 3 lines of page 3 (the second manuscript page)

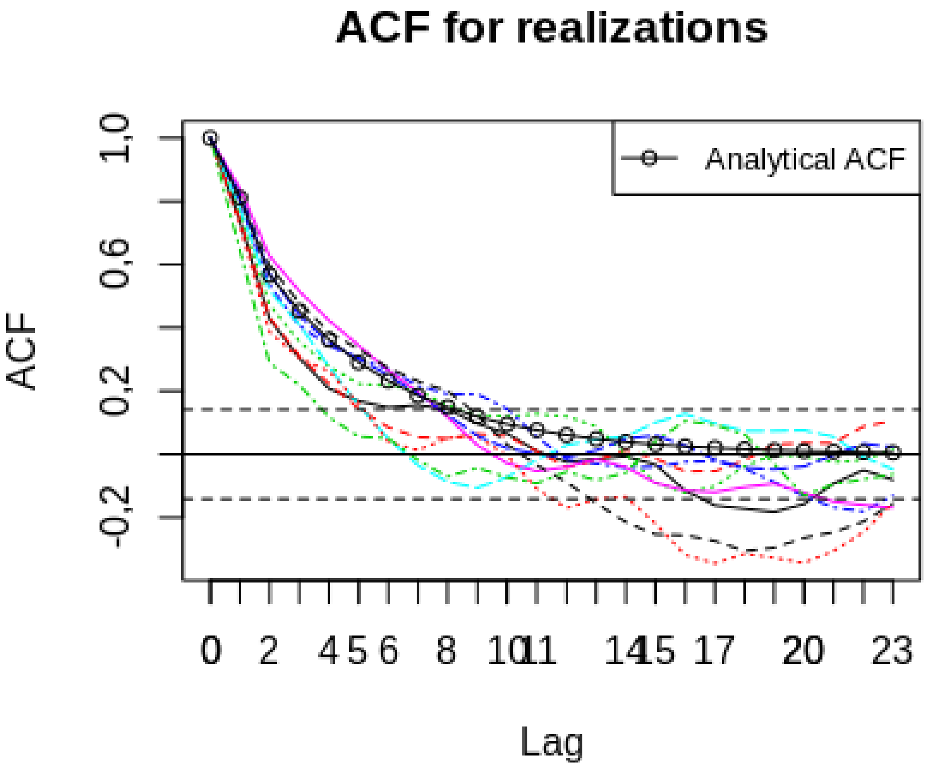


Figure 5: ACF of 10 realizations + Theoretical ACF

The estimated ACF for the 10 realizations follows the theoritical ACF in the behaviour :the way of decreasing towards 0 and the corresponding number of lags. However, there are some differences because they do not fit exactly the same.

2 Question 2.2

The quarterly number of sales of apartments in the capital region of Denmark has been modelled. Based on historical data the following model has been identified where ϵ_t is a white-noise process with variance $\sigma_\epsilon^2 = 0.36936$:

$$(1 - 0.99B + 0.2B^2)(1 - 0.62B^4)(Y_t - \mu) = \epsilon_t \tag{5}$$

First of all, it always good to plot the data. Then the process should be verified whether it is invertible and stationary.
photo
The process $_t$ = Y_t - μ is said to follow a multiplicative (p=2,d=0,q=0)*(P=1,D=0,Q=0) seasonal model.
According to the definition (5.22) p132 and according to the A2_sales.txt document that has

4 observations each year, the time series described has s=4. Hence it is called the quarterly sales. The process can be written :

$$\phi(B)\Phi(B^4)(Z_t) = \epsilon_t \tag{6}$$

The process is stationary because d=D=0 according to p132 of the book. It is invertible because q=0 .

The prediction is obtained by the following lines using p137:
Let t be the current number of observations i.e t = 20. The task is to predict the next two steps meaning calculating $\hat{Y}_{t+1}$ and $\hat{Y}_{t+2}$. For now the predictions will be made on $\hat{Z}_{t+1}$ and $\hat{Z}_{t+2}$ As know, the best predictor is :

$$\hat{Z}_{t+k|t} = E[Z_{t+k}|Z_t, Z_{t-1}, ...] \tag{7}$$

The formula (6) can be expanded :

$$(1 - 0.62B^4 - 0.99B + 0.6138B^5 + 0.22B^2 - 0.1364B^6)Z_t = \epsilon_t \tag{8}$$

which can be written as :

$$Z_t = 0.99Z_{t-1} - 0.22Z_{t-2} + 0.62Z_{t-4} - 0.6138Z_{t-5} + 0.1364Z_{t-6} + \epsilon_t \tag{9}$$

Using (7), the assumption that ϵ_t is white noise which gives

$$E[\epsilon_t|Z_t, Z_{t-1}, Z_{t-2}...] = 0 \tag{10}$$

and that the observations are known, so they are just constants, it can be concluded that :

$$\hat{Z}_{t+1|t} = 0.99Z_t - 0.22Z_{t-1} + 0.62Z_{t-3} - 0.6138Z_{t-4} + 0.1364Z_{t-5} \tag{11}$$

$$\hat{Z}_{21|20} = 0.99Z_{20} - 0.22Z_{19} + 0.62Z_{17} - 0.6138Z_{16} + 0.1364Z_{15} \tag{12}$$

And:

$$\hat{Z}_{t+2|t} = 0.99\hat{Z}_{t+1|t} - 0.22Z_t + 0.62Z_{t-2} - 0.6138Z_{t-3} + 0.1364Z_{t-4} \tag{13}$$

$$\hat{Z}_{22|20} = 0.99\hat{Z}_{21|20} - 0.22Z_{20} + 0.62Z_{18} - 0.6138Z_{17} + 0.1364Z_{16} \tag{14}$$

The goal is Y_t so :

$$\hat{Y}_{21|20} = \hat{Z}_{21|20} + \mu = 2096,963 \tag{15}$$

$$\hat{Y}_{22|20} = \hat{Z}_{22|20} + \mu = 2159,353 \tag{16}$$

The confidence interval can be found using p137,(5.149),(5.151) and remark 5.5 p138 :

$$e_{t+k|t} = Y_{t+k} - \hat{Y}_{t+k|t} \tag{17}$$

$$\hat{Y}_{t+k|t} \pm u_{\alpha/2}\sqrt{\text{Variance}(e_{t+k|t})} \tag{18}$$

where $u_{\alpha/2}$ is the $\alpha/2$ quantile in the standard normal distribution.
According to (5.151),
For k=1, $\psi_0 = 1$

$$variance((e_{t+1|t}) = \sigma_\epsilon^2(1) \tag{19}$$

For k=2, $\psi_0 = 1$, $\psi_1 = 0.99$ ie

$$variance((e_{t+2|t}) = \sigma_\epsilon^2(1 + 0.99^2) \tag{20}$$

Finally with choosing $\alpha = 0.05$ giving 95% confidence interval :

$$\hat{Y}_{t+1|t} \pm u_{\alpha/2}\sqrt{\text{Variance}(e_{t+1|t})} = 2096.963 \pm 389.5746 \tag{21}$$

$$\hat{Y}_{t+2|t} \pm u_{\alpha/2}\sqrt{\text{Variance}(e_{t+2|t})} = 2159.35 \pm 548.1939 \tag{22}$$

$$\tag{23}$$

where (t+1) correspond to 2019Q1 and (t+2) to 2019Q2 Due to high $\sigma_{epsilon}$ the confidence interval are large and they get larger when predicting further steps as explained in (5.151).

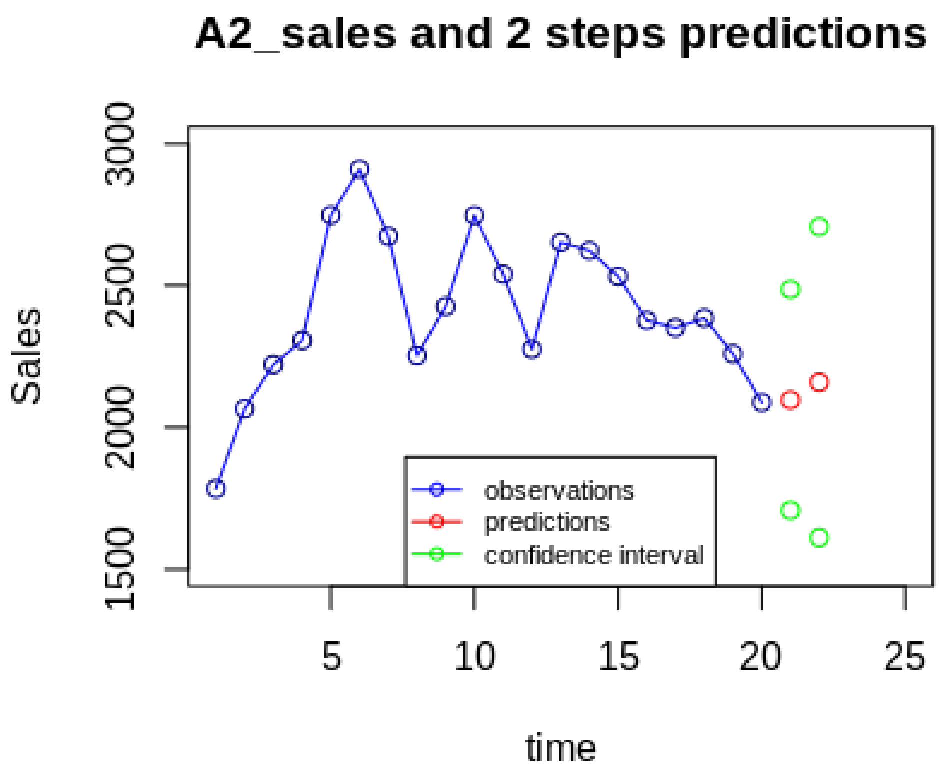


Figure 6: A2_sales and 2 steps predictions with confidence intervals

3 Question 2.3

1)The following ARMA(2,0) is given:

$$\phi(B)X_t = \epsilon_t \tag{24}$$

$$(1 - 1.5B + \phi_2 B^2)X_t = \epsilon_t \tag{25}$$

Multiplying the equation by z^2

$$\phi(z^{-1}) = 0 \equiv 1 + 1.5z + \phi_2 z^2 = 0 \tag{26}$$

For ϕ_2=0.52 : roots = [-0,5438447+0i ; 0,9561553-0i] which are real and <1 the process is then stationary
For ϕ_2=0.98 : roots = [-0,75+0,6461424i ; -0,75-0,6461424i] giving a norm = 0,9899495 <1 the process is then stationary
2)There are 4 processes:
Process 1: ϕ_2=0.52 , σ=0.1
Process 2: ϕ_2=0.52 , σ=5
Process 3: ϕ_2=0.98 , σ=0.1
Process 4: ϕ_2=0.98 , σ=5
To estimate parameters,the function arima() in R is used. This function uses the maximum likelihood ML method by default.
set.seed(123) was used in R in order to generate the same realizations.
The red line indicated the 95% quantiles (which are the 2.5% and the 97.5% quantiles)

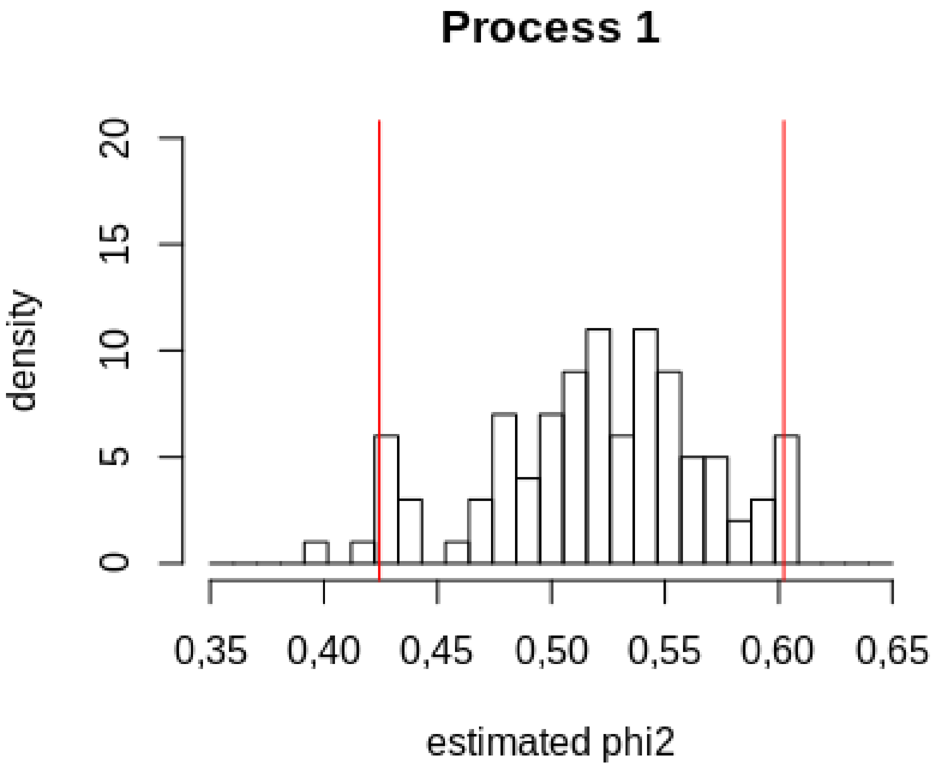


Figure 7: Process 1

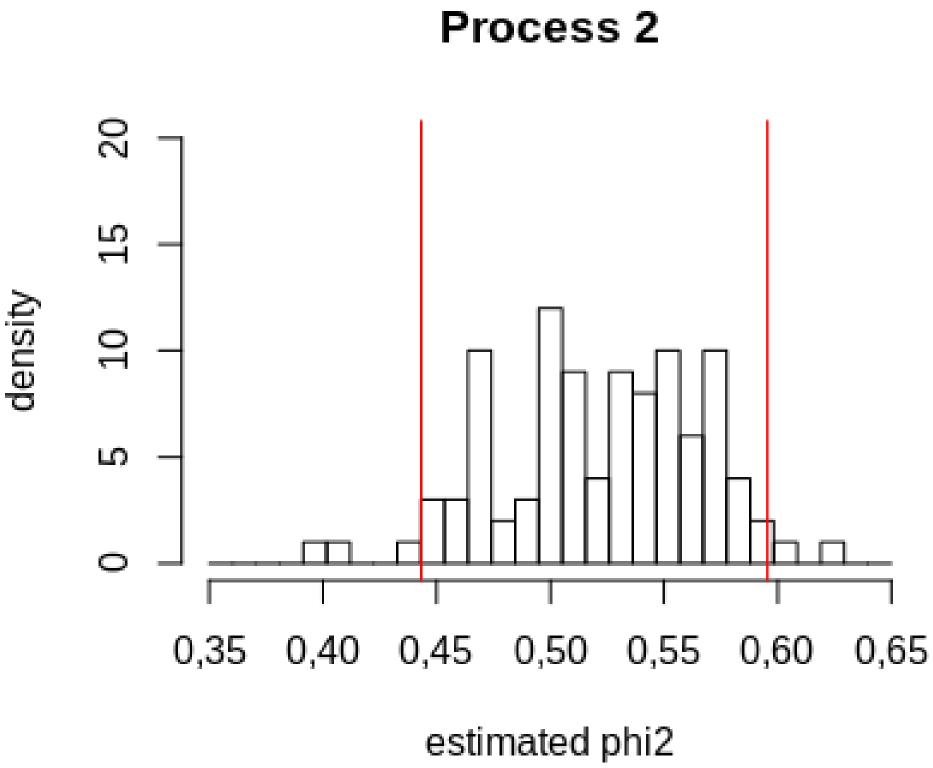


Figure 8: Process 2

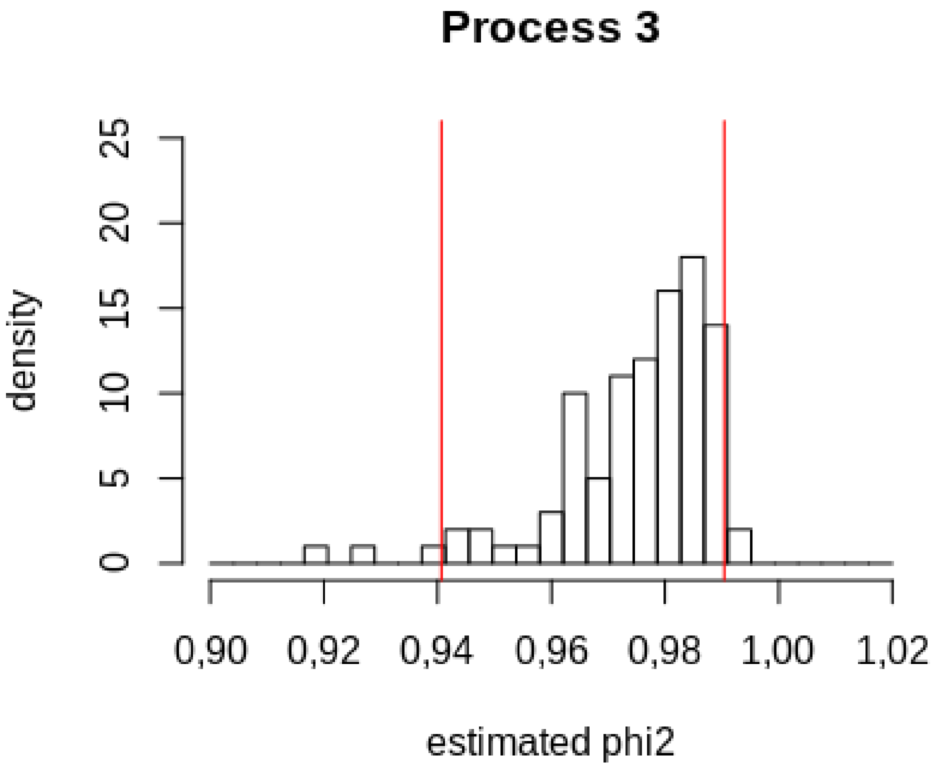


Figure 9: Process 3

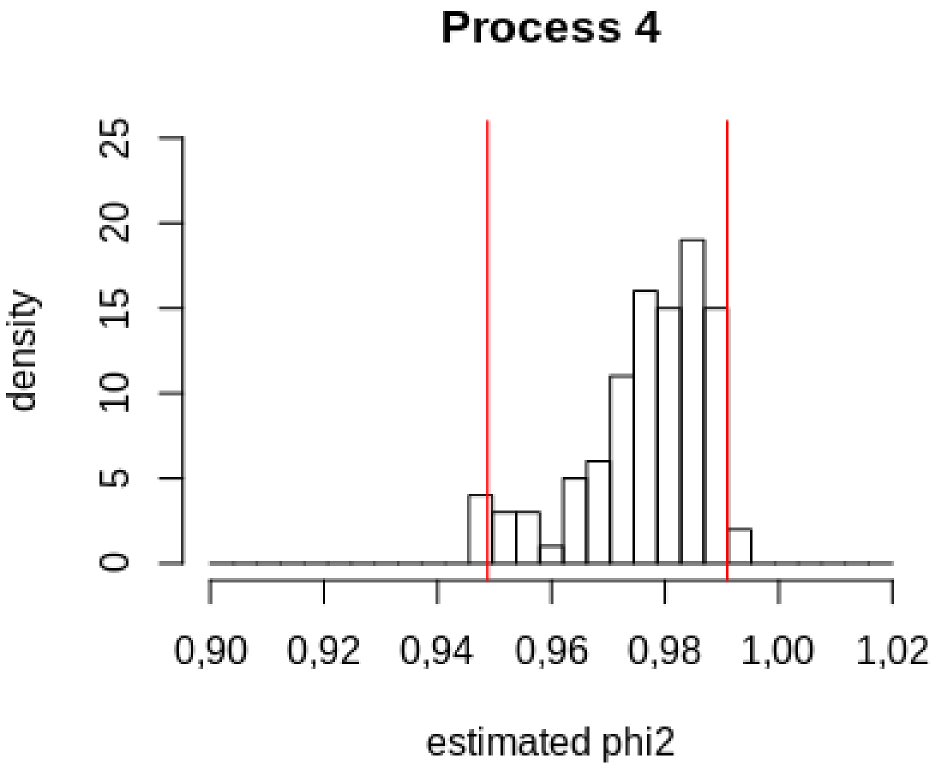


Figure 10: Process 4

3)When ϕ_2 varies, the distribution of the estimated ϕ_2 varies.
4)It seems that the different values of the noise σ does not affect the distribution of ϕ_2
3-4)One way to validate more the conclusion above is trying to fit the distribution of ϕ_2 by the normal distribution and conclude the same results as before.
Using the library "MASS" in R, the estimated ϕ_2 were fitted to a normal distribution. The results given by this library are not perfect as seen in the plots but prove the same point discussed.

$$Process1: \quad \hat{\phi}_2 \sim \mathcal{N}(0.52, 0.0023) \tag{27}$$

$$Process2: \quad \hat{\phi}_2 \sim \mathcal{N}(0.52, 0.002) \tag{28}$$

$$Process3: \quad \hat{\phi}_2 \sim \mathcal{N}(0.97, 0.0, 00019) \tag{29}$$

$$Process4: \quad \hat{\phi}_2 \sim \mathcal{N}(0.97, 0.00012) \tag{30}$$

This proves that σ does not affect the variance/distribution of the estimated ϕ_2 since the variances in process1 and process2 is very close and yet these processes have different σ (same for process3 and process4) .

This also proves that ϕ_2 affect the variance/distribution of the estimated ϕ_2 because when ϕ_2 varies from a process to another, the distribution and the variance change.
To run the code in this section,the last part in the code should be executed in order to have the green normal curves.

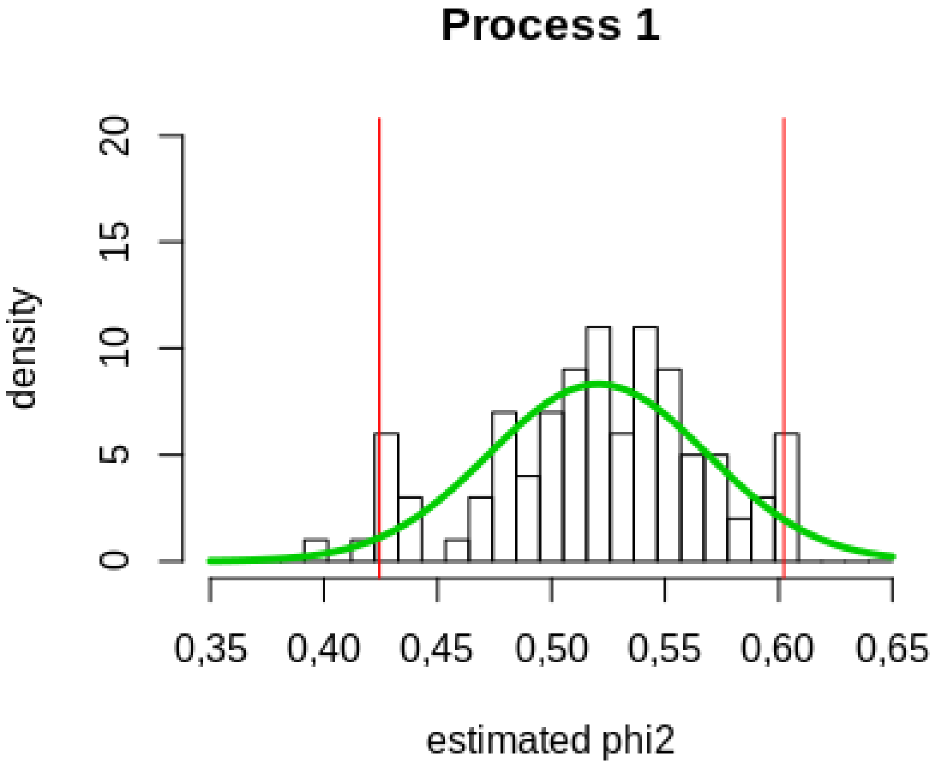


Figure 11: Process 1

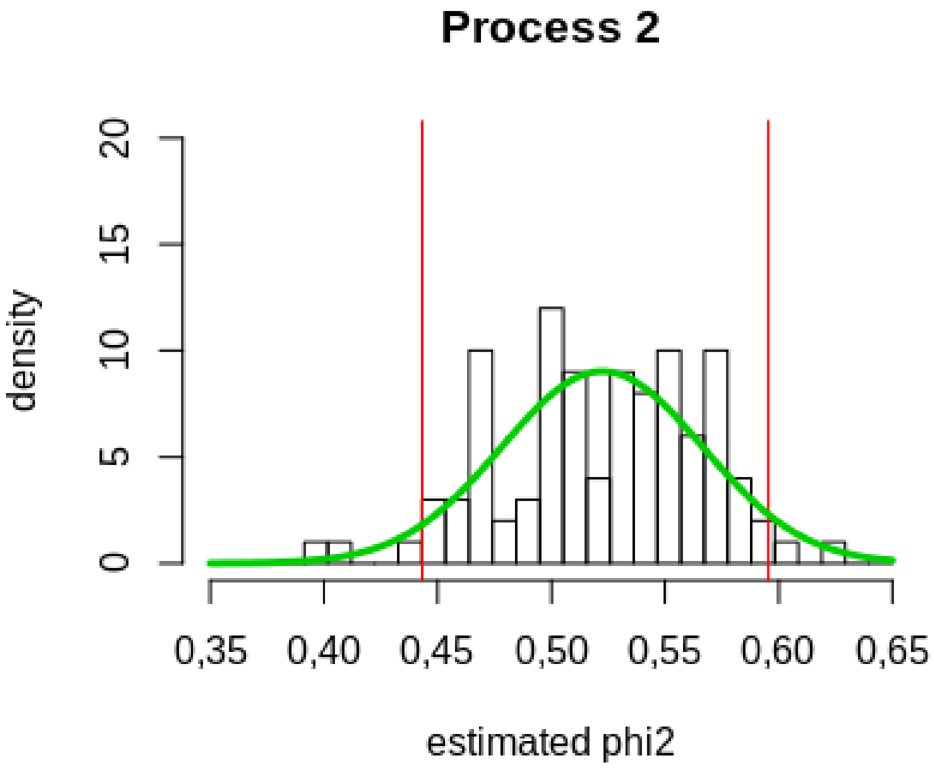


Figure 12: Process 2

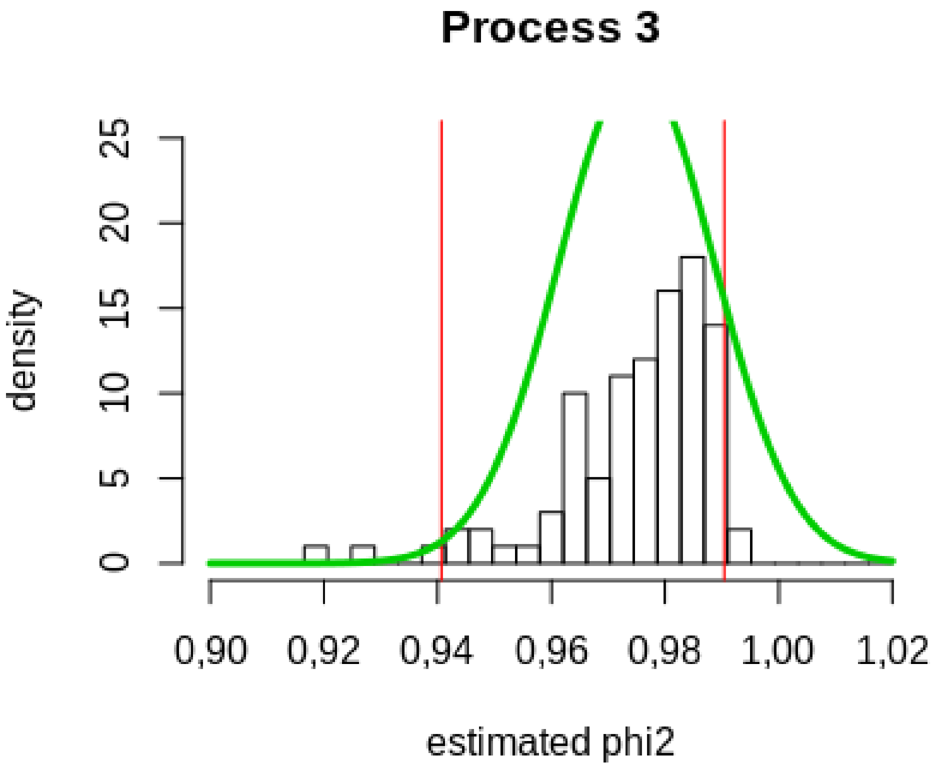


Figure 13: Process 3

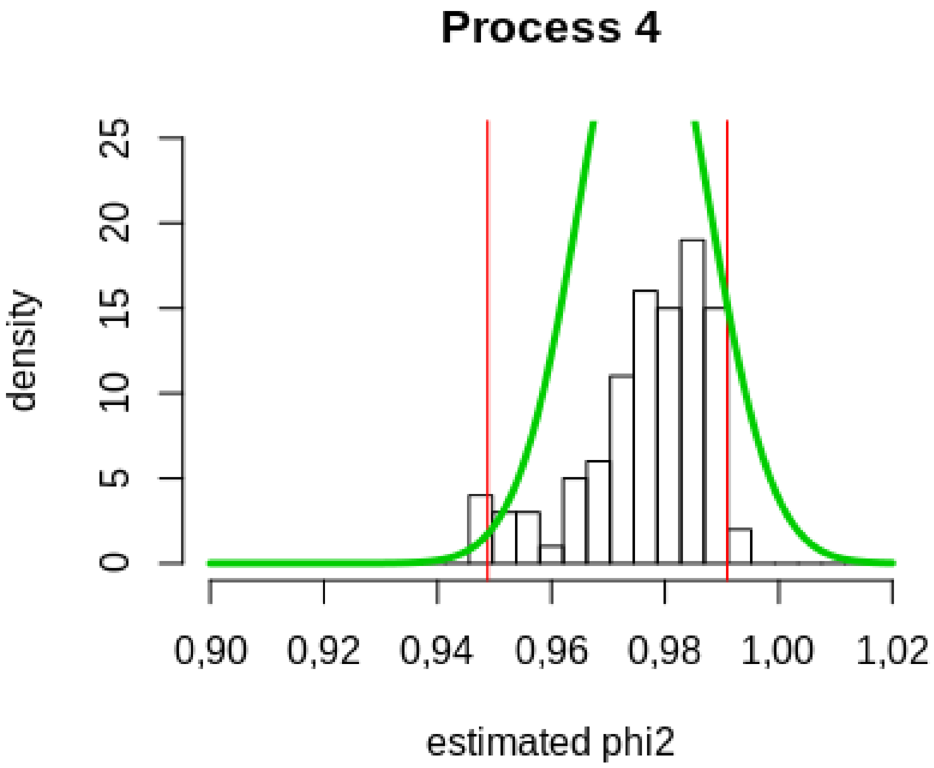


Figure 14: Process 4

5)

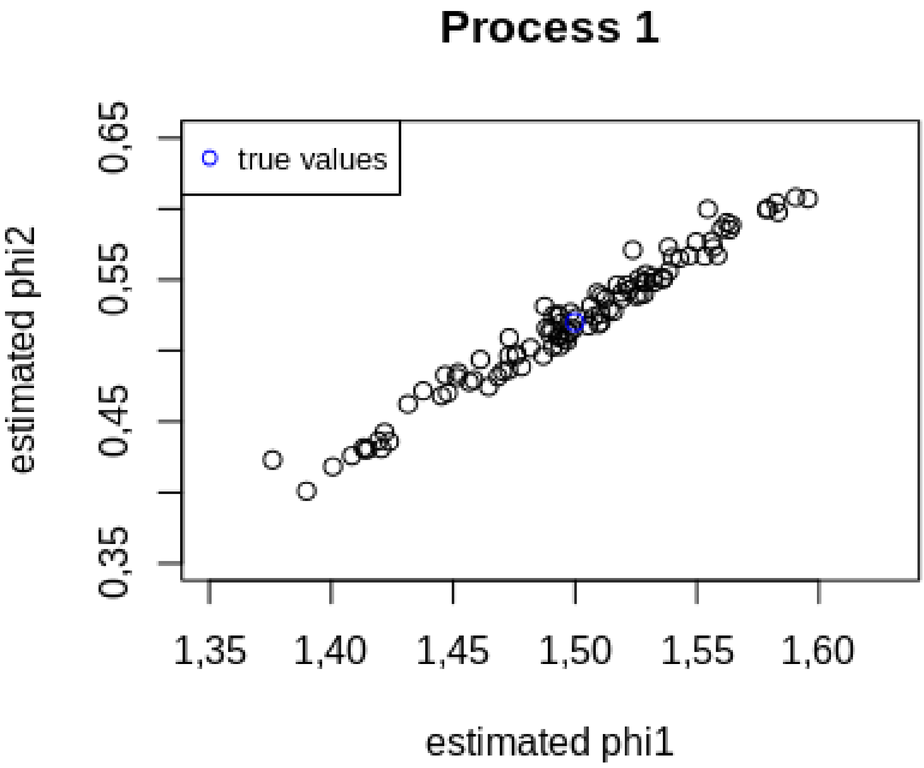


Figure 15: Process 1

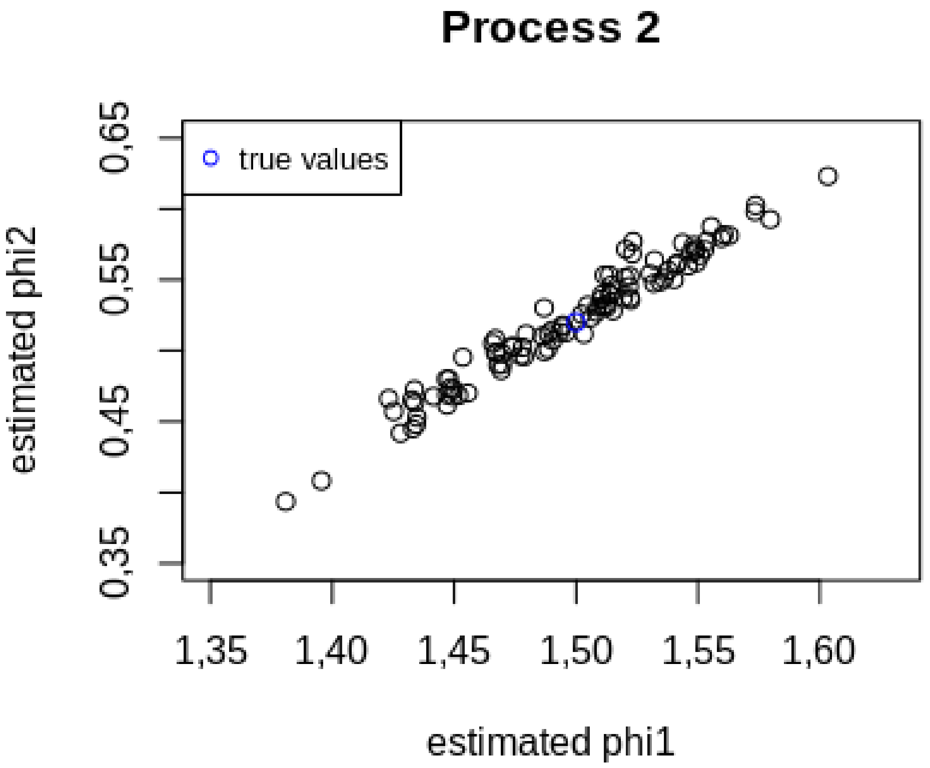


Figure 16: Process 2

It can be seen that from these two plots,which differ by the variance, that the variance in the simulations is not important.

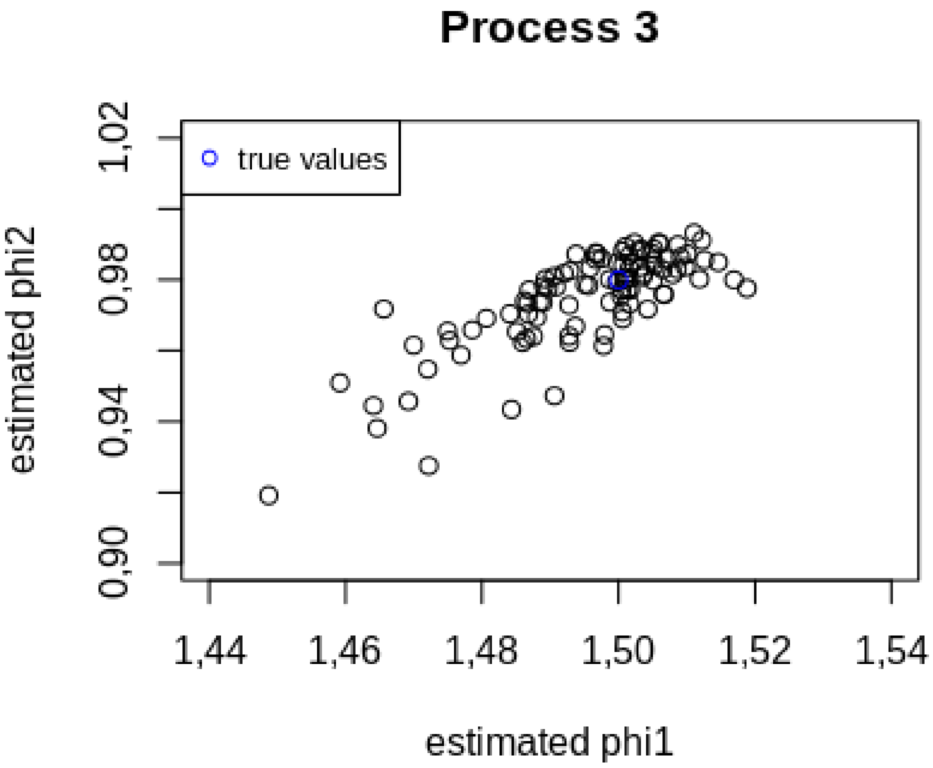


Figure 17: Process 3

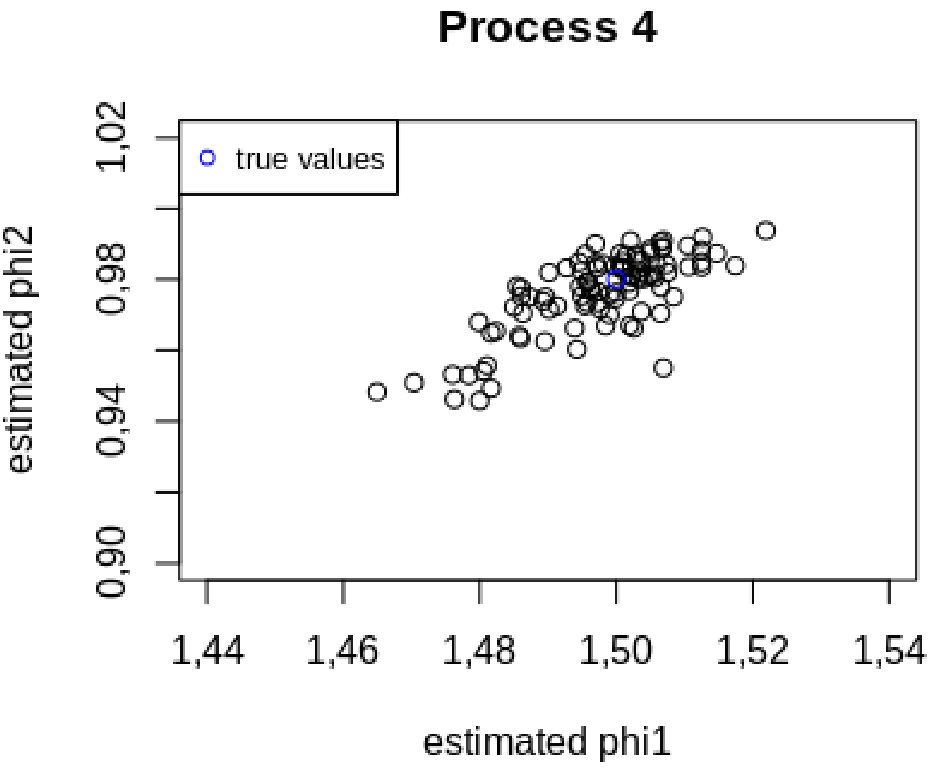


Figure 18: Process 4

It can be seen that from these two plots,which differ by the variance, that the variance in the simulations is not important.
The estimated values of ϕ_1 and ϕ_2 are close to the real values.
One can ask how are ϕ_1 and ϕ_2 correlated after seeing these plots. Using the correlation function " cor() " and the covariance function " cov() " in R. The variance is calculated using "var()" :
-Process1: cor(ϕ_1,ϕ_2)= ,0,98365 and cov(ϕ_1,ϕ_2)=0,002272615 and var(ϕ_1)=0,002297662 and var(ϕ_2)=0,002323189
-Process2: cor(ϕ_1,ϕ_2)= 0,9777678 and cov(ϕ_1,ϕ_2)= 0,001920491 and var(ϕ_1)=0,001954405 and var(ϕ_2)=0,001973961
This also proves that the variance of the noise has no importance in the simulations since the correlations and covariances of process1 and process2 are close.
-Process3: cor(ϕ_1,ϕ_2)=0,7901635 and cov(ϕ_1,ϕ_2)= 0,0001450862 and var(ϕ_1)=0,0001767456 and var(ϕ_2)=0,0001907524
-Process4: cor(ϕ_1,ϕ_2)=0,7532301 and cov(ϕ_1,ϕ_2)= 8,837276e-05 and var(ϕ_1)=0,0001092191 and var(ϕ_2)=0,0001260326
This also proves that the variance of the noise has no importance in the simulations since the correlations and covariances of process3 and process4 are close.

It can also be seen from variances values and the histograms above that the variance of the estimated ϕ_1 and ϕ_2 increases when the correlation between them increases.

Since in the beginning, it is said that the process is stationary, one can hope that the for the estimated values of ϕ_2 and ϕ_2 all the 4 processes are stationary. This needs to be verified using like before the roots of $\phi(z^{-1})$ However, just by typing " norm_roots_process1 " or the process wanted, many solutions has roots > 1, saying that that realization of process for the process is not stationary.

4 Code

```

rm(list=ls())

#Question2.1####################################################################################
##1)#####################################################################################
polyroot(c(-0.3,0.4,1))
##2)see the report
##3)
###The phi_i parameters in the 'arima' function in R are positive on the right hand side of the equal sign
###so we implement 0.8 and not (-0.8) in AR part
N<-200
set.seed(123)
sim = replicate(10, arima.sim(list(order = c(1,0,2), ar = c(0.8), ma = c(0.4, -0.3)),n = 200, sd = 0.4))
matplot(sim, type = "l", ylab="X_t", xlab="t")
##4)
simulation_acf = sapply(split(sim,col(sim)), function(ts) acf(ts,plot=FALSE)$acf)
x_seq = seq(from = 0, to=length(simulation_acf[,1])-1, by = 1)
matplot(x_seq, simulation_acf, type="l",ylab="ACF", xlab="Lag", main = "ACF for realizations")
axis(1, at=x_seq)
abline( 2/sqrt(N),0, lwd=1, lty=2)
abline(-2/sqrt(N),0, lwd=1, lty=2)

##5)
```

```
simulation_pacf = sapply(split(sim,col(sim)), function(ts) pacf(ts,plot=FALSE)$acf)
x_seq_2 = seq(from = 0, to=length(simulation_pacf[,1]), by = 1)
matplot(simulation_pacf, type="l",ylab="PACF", xlab="Lag",  main = "PACF for realizations")
axis(1,at=x_seq_2)
abline(2/sqrt(N),0, lwd=1, lty=2)
abline(-2/sqrt(N),0, lwd=1, lty=2)
##6)
#sim_var = sapply(split(sim,col(sim)), function(ts) var(ts))
sim_acof = sapply(split(sim,col(sim)), function(ts) acf(ts,plot=F, type="covariance")$acf)
plot(sim_acof[1,],ylab="Variance[[X(FFFD)w] ]", xlab = " Realization w ")
##7)
### Theoritical ACF autocorrelation function ###
rho_0<- 1
rho_1<- 0.81096
rho_2<- 0.56657
rho_list <- 3:23
rho<- c(rep(0,length(x_seq))  )
for (i in rho_list)
{
  rho[i+1]<- (0.8)^( i -2) * rho_2   # rho[i] here = rho[i-1] in manuscript
}
rho[1]<-rho_0
rho[2]<-rho_1
rho[3]<-rho_2

matplot(x_seq, simulation_acf, type="l",ylab="ACF", xlab="Lag", main = "ACF for realizations")
axis(1, at=x_seq)
abline( 2/sqrt(N),0, lwd=1, lty=2)
abline(-2/sqrt(N),0, lwd=1, lty=2)
abline(0,0, lwd=1, lty=1)
lines(x_seq,rho, type="l",lty=1, col="black")
points(x_seq,rho, cex=0.8 , col="black")
legend("topright", legend=c("Analytical ACF"),  col=c("black"), pch=c(1), pt.bg =c("black") ,lty=1, cex=0.8 )

#Question2.2##################################################################################
A2 = read.table("/home/ghassen97/Desktop/S8/time series analysis/assignment/assignment 2/A2_sales.txt", header = TRUE)
```

```
plot(A2$Sales, ylab="Sales", xlab='time', main="A2_sales", type = "l")
points(A2$Sales, cex = .5, col = "dark blue")

#predicting 2 steps ahead:#
mu=2092

Z= A2$Sales - mu  # variable change
t=21
Z[21] <- 0.99*Z[t-1]-0.22*Z[t-2]+0.62*Z[t-4]-0.6138*Z[t-5]+0.1364*Z[t-6]
t=22
Z[22] <- 0.99*Z[t-1]-0.22*Z[t-2]+0.62*Z[t-4]-0.6138*Z[t-5]+0.1364*Z[t-6]

Y= Z+ mu
plot(Y[1:20], type = 'l', xlim = c(1,25), ylab ='time', main = 'A2_sales and
     2 steps predictions',col="blue")
points(A2$Sales, col = "dark blue")
points(21, Y[21], col = 'red')
points(22, Y[22], col = 'red')
legend("topright", legend=c("observations", "predictions"), col=c("blue", "
   red"),pch=c(1,1), pt.bg =c("blue", "red") ,lty=1, cex=0.7)

#Confidence intervals#
k<- 1
alpha<-0.05
sigma_e<-sqrt(39508)
half_interval_t1 = qt(1-alpha/2, df=Inf ) * sigma_e*sqrt(1)
half_interval_t2 = qt(1-alpha/2, df=Inf ) * sigma_e*sqrt(1+0.99^2)
plot(Y[1:20], type = 'l', xlim = c(1,25),xlab='time', ylab ='Sales', ylim=c
   (1500,3000) , main = 'A2_sales and 2 steps predictions',col="blue")
points(A2$Sales, col = "dark blue")
points(21, Y[21], col = 'red')
points(21, Y[21]+half_interval_t1 , col="green" )
points(21, Y[21]-half_interval_t1 , col="green" )
points(22, Y[22], col = 'red')
points(22, Y[22]+half_interval_t2 , col="green" )
points(22, Y[22]-half_interval_t2 , col="green" )
legend("bottom", legend=c("observations", "predictions","confidence interval
   "), col=c("blue", "red","green"),pch=c(1,1,1), pt.bg =c("blue", "red","
   green") ,lty=1, cex=0.7)

#Question
   2.3#################################################################################

##1)
```

```r
phi2_1<- 0.52
phi2_2<- 0.98
roots_1 <- polyroot(c(phi2_1, 1.5 , 1))
norm_roots_1<- Mod(roots_1)
roots_2 <- polyroot(c(phi2_2, 1.5 , 1))
norm_roots_2<- Mod(roots_2)
##2)
###process1###
set.seed(123)
phi2_ind<-2
break1 <- seq(0.35, 0.65, length.out = 30) # to have same number of bids, and adjust x_axis

sim1 <- replicate(100, arima.sim(list(order = c(2,0,0), ar = c(-1.5, -0.52)), n = 300, sd = 0.1))
sim1_arima_process1 <- sapply(split(sim1,col(sim1)), function(ts) arima(ts, order = c(2,0,0))$coef)*(-1)
sim1_arima <- sim1_arima_process1[phi2_ind,]
sim1_arima_histogram<-hist(sim1_arima,breaks=break1,plot=FALSE)
plot(sim1_arima_histogram, xlab='estimated phi2', ylab='density', ylim = c(0,20), main="Process 1" )

p1<-quantile(sim1_arima,0.975)
p2<-quantile(sim1_arima,0.025)

abline(v=p1, col='red')
abline(v=p2,col='red')

curve(dnorm(x, mean=0.52061432, sd=0.04795787),add = TRUE,col=3,lwd=3)


###process2###
break2<-break1

sim2 <- replicate(100, arima.sim(list(order = c(2,0,0), ar = c(-1.5, -0.52)), n = 300, sd = 5))
sim2_arima_process2 <- sapply(split(sim2,col(sim2)), function(ts) arima(ts, order = c(2,0,0))$coef)*(-1)
sim2_arima <- sim2_arima_process2[phi2_ind,]

sim2_arima_histogram<-hist(sim2_arima,breaks=break2,plot=FALSE)
plot(sim2_arima_histogram, xlab='estimated phi2', ylab='density', ylim = c(0,20),main="Process 2")
p2_1<-quantile(sim2_arima,0.975)
```

```r
p2_2<-quantile(sim2_arima,0.025)
abline(v=p2_1, col='red')
abline(v=p2_2,col='red')

curve(dnorm(x, mean=0.522379960, sd=0.044206581),add = TRUE,col=3,lwd=3)


###process 3#####
break3 = seq(0.9, 1.02, length.out = 30)

sim3 <- replicate(100, arima.sim(list(order = c(2,0,0), ar = c(-1.5, -0.98)), n = 300, sd = 0.1))
sim3_arima_process3 <- sapply(split(sim3,col(sim3)), function(ts) arima(ts, order = c(2,0,0))$coef)*(-1)
sim3_arima <- sim3_arima_process3[phi2_ind,] # phi2

sim3_arima_histogram<-hist(sim3_arima,breaks = break3,plot=FALSE)
plot(sim3_arima_histogram, xlab='estimated phi2', ylab='density', ylim = c(0,25),main="Process 3")
p3_1<-quantile(sim3_arima,0.975)
p3_2<-quantile(sim3_arima,0.025)
abline(v=p3_1, col='red')
abline(v=p3_2,col='red')

curve(dnorm(x, mean=0.974997395, sd=0.013742083),add = TRUE,col=3,lwd=3)


###process 4#####
break4<-break3
sim4 <- replicate(100, arima.sim(list(order = c(2,0,0), ar = c(-1.5, -0.98)), n = 300, sd = 5))
sim4_arima_process4 <- sapply(split(sim4,col(sim4)), function(ts) arima(ts, order = c(2,0,0))$coef)*(-1)
sim4_arima <- sim4_arima_process4[phi2_ind,] # phi2

sim4_arima_histogram<-hist(sim4_arima,breaks = break4,plot=FALSE)
plot(sim4_arima_histogram, xlab='estimated phi2', ylab='density', ylim = c(0,25),main="Process 4")
p4_1<-quantile(sim4_arima,0.975)
p4_2<-quantile(sim4_arima,0.025)
abline(v=p4_1, col='red')
abline(v=p4_2,col='red')

curve(dnorm(x, mean=0.976334331, sd=0.011170152),add = TRUE,col=3,lwd=3)
```

```r


#3) #4) see report
#5)
### Process 1 ###
phi1_1_hat<-sim1_arima_process1[1,]
phi2_1_hat<-sim1_arima_process1[2,]
plot(phi1_1_hat,phi2_1_hat,ylim=c(0.35,0.65), xlim=c(1.35, 1.63), xlab="estimated phi1", ylab = "estimated phi2", main='Process 1')
points(1.5, 0.52, col="blue") #true value
legend("topleft", legend=c("true values"),  col=c("blue"), pch=c(1), pt.bg = c("blue"), cex=0.8 )

cov1<-cov(phi1_1_hat,phi2_1_hat,use = "everything",method = "pearson")
cor1<-cor(phi1_1_hat,phi2_1_hat,use = "everything",method = "pearson")

var(phi1_1_hat)
var(phi2_1_hat)

### checking stationarity for all phi1 and phi2 ###
roots_process1 <- polyroot(c(phi2_1_hat, phi1_1_hat , 1))
norm_roots_process1<- Mod(roots_process1)


### Process 2 ###
phi1_2_hat<-sim2_arima_process2[1,]
phi2_2_hat<-sim2_arima_process2[2,]
plot(phi1_2_hat,phi2_2_hat,ylim=c(0.35,0.65), xlim=c(1.35, 1.63),xlab="estimated phi1", ylab = "estimated phi2", main='Process 2')
points(1.5, 0.52, col="blue")
legend("topleft", legend=c("true values"),  col=c("blue"), pch=c(1), pt.bg = c("blue"), cex=0.8 )

cov2<-cov(phi1_2_hat,phi2_2_hat,use = "everything",method = "pearson")
cor2<-cor(phi1_2_hat,phi2_2_hat,use = "everything",method = "pearson")

var(phi1_2_hat)
var(phi2_2_hat)
### checking stationarity for all phi1 and phi2 ###
roots_process2 <- polyroot(c(phi2_2_hat, phi1_2_hat , 1))
norm_roots_process2<- Mod(roots_process2)

### Process 3 ###
phi1_3_hat<-sim3_arima_process3[1,]
```

```
phi2_3_hat<-sim3_arima_process3[2,]
plot(phi1_3_hat,phi2_3_hat,ylim=c(0.9,1.02), xlim=c(1.44, 1.54),xlab="
   estimated phi1", ylab = "estimated phi2", main='Process 3')
points(1.5, 0.98, col="blue")
legend("topleft", legend=c("true values"),  col=c("blue"), pch=c(1), pt.bg =
   c("blue"), cex=0.8 )

cov3<-cov(phi1_3_hat,phi2_3_hat,use = "everything",method = "pearson")
cor3<-cor(phi1_3_hat,phi2_3_hat,use = "everything",method = "pearson")

var(phi1_3_hat)
var(phi2_3_hat)
### checking stationarity for all phi1 and phi2 ###
roots_process3 <- polyroot(c(phi2_3_hat, phi1_3_hat , 1))
norm_roots_process3<- Mod(roots_process3)

### Process 4 ###
phi1_4_hat<-sim4_arima_process4[1,]
phi2_4_hat<-sim4_arima_process4[2,]
plot(phi1_4_hat,phi2_4_hat,ylim=c(0.9,1.02), xlim=c(1.44, 1.54),xlab="
   estimated phi1", ylab = "estimated phi2", main='Process 4')
points(1.5, 0.98, col="blue")
legend("topleft", legend=c("true values"),  col=c("blue"), pch=c(1), pt.bg =
   c("blue"), cex=0.8 )

cov4<-cov(phi1_4_hat,phi2_4_hat,use = "everything",method = "pearson")
cor4<-cor(phi1_4_hat,phi2_4_hat,use = "everything",method = "pearson")

var(phi1_4_hat)
var(phi2_4_hat)
### checking stationarity for all phi1 and phi2 ###
roots_process4 <- polyroot(c(phi2_4_hat, phi1_4_hat , 1))
norm_roots_process4<- Mod(roots_process4)


### fitting histograms with normal distribution ### to be executed before
### the 4 lines like:  curve(dnorm(x, mean=0.976334331, sd=0.011170152),add
   = TRUE,col=3,lwd=3)

library("MASS")
a4<-fitdistr(phi2_4_hat, "normal")
a3<-fitdistr(phi2_3_hat, "normal")
a2<-fitdistr(phi2_2_hat, "normal")
a1<-fitdistr(phi2_1_hat, "normal")
```

List of Figures