Danmarks Tekniske Universitet

DTU Compute - Institut for Matematik og Computerscience

Assignment 2: ARMA and Seasonal Processes

Christian Adam Deding Nielsen - s204261
Emilie Østerdal Nilsson - s204204
Magnus Holm - s204245
Oscar Bach Østdahl Pedersen - s204243

02417 Time Series Analysis Spring 23

March 2023

Question 2.1

2.1.1

Using definition 5.19 we see the process is an ARMA(1,2) process and we can thus write it as

$$\phi(B)X_t = \theta(B)\varepsilon_t$$

where the polynomials ϕ and θ are

$$\begin{aligned}\phi(B) &= 1 - 0.8B \\ \theta(B) &= 1 + 0.8B - 0.5B^2.\end{aligned}$$

The root of $\phi(z^{-1}) = 1 - 0.8/z$ is $z = 0.8$ which is inside the unit circle, hence the process is stationary by theorem 5.12. The roots of $\theta(z^{-1}) = 1 + 0.8/z - 0.5/z^2$ are $z \approx 0.41$ and $z \approx -1.21$. As -1.21 is not within the unit circle the process is not invertible.

2.1.2

For the second order moment representation we aim to find the mean value and the autocovariance function of the process. Setting $a = 0.8$ and $b = 0.5$ the process X_t becomes

$$X_t = aX_{t-1} + \varepsilon_t + a\varepsilon_{t-1} - b\varepsilon_{t-2} \tag{1}$$

By successively substituting $X_{t-1} = aX_{t-2}+\varepsilon_{t-1}+a\varepsilon_{t-2}-b\varepsilon_{t-3}$, $X_{t-2} = aX_{t-3}+\varepsilon_{t-2}+a\varepsilon_{t-3}-b\varepsilon_{t-4}$ we see that

$$\begin{aligned}X_t &= a(aX_{t-2} + \varepsilon_{t-1} + a\varepsilon_{t-2} - b\varepsilon_{t-3}) + \varepsilon_t + a\varepsilon_{t-1} - b\varepsilon_{t-2} \\ &= a^2X_{t-2} + 2a\varepsilon_{t-1} + (a^2 - b)\varepsilon_{t-2} - ab\varepsilon_{t-3} + \varepsilon_t \\ &= a^2(aX_{t-3} + \varepsilon_{t-2} + a\varepsilon_{t-3} - b\varepsilon_{t-4}) + 2a\varepsilon_{t-1} + (a^2 - b)\varepsilon_{t-2} - ab\varepsilon_{t-3} + \varepsilon_t \\ &= a^3X_{t-3} + (2a^2 - b)\varepsilon_{t-2} + (a^3 - ab)\varepsilon_{t-3} - a^2b\varepsilon_{t-4} + 2a\varepsilon_{t-1} + \varepsilon_t \\ &= a^3(aX_{t-4} + \varepsilon_{t-3} + a\varepsilon_{t-4} - b\varepsilon_{t-5}) + (2a^2 - b)\varepsilon_{t-2} + (a^3 - ab)\varepsilon_{t-3} - a^2b\varepsilon_{t-4} + 2a\varepsilon_{t-1} + \varepsilon_t \\ &= a^4X_{t-4} - a^3b\varepsilon_{t-5} + (a^4 - a^2b)\varepsilon_{t-4} + (2a^3 - ab)\varepsilon_{t-3} + (2a^2 - b)\varepsilon_{t-2} + 2a\varepsilon_{t-1} + \varepsilon_t\end{aligned}$$

and by continuing in this fashion we see that the process is linear and can be expressed in the following way

$$X_t = (\sum_{i=0}^{\infty} \psi_i B^i)\varepsilon$$

where

$$\psi_0 = 1,\ \psi_1 = 2a = 1.6,\ \psi_i = 2a^i - a^{i-2}b,\ i = 2,3,4,...$$

Given that $\mathrm{E}[\varepsilon_t] = 0$, we thus find that

$$\mu_X = \mathrm{E}[X_t] = 0$$

From the polynomials ϕ and θ we obtain the coefficients

$$\phi_0 = 1,\, \phi_1 = -0.8$$

and

$$\theta_0 = 1,\; \theta_1 = 0.8\; \theta_2 = -0.5.$$

We define $\theta_k = 0$ for $k \notin \{0,1,2\}$. Using eq. (5.97) in the book we determine $\gamma_{\varepsilon Y}$ as defined in eq. (5.96) in the book, by using and $\sigma_\varepsilon = 0.4$:

$$k = 0: \quad \gamma_{\varepsilon Y}(0) = \theta_0 \sigma_\varepsilon^2 = 0.16$$

$$k \in \mathbb{N}: \quad \gamma_{\varepsilon Y}(k) = \theta_k \sigma_\varepsilon^2 + \phi_1 \gamma_{\varepsilon Y}(k-1)$$

Where k is the lag. We can calculate values

$$\gamma_{\varepsilon Y}(1) = 0.8 \cdot 0.16 - 0.8 \cdot 0.16 = 0$$

and

$$\gamma_{\varepsilon Y}(2) = -0.5 \cdot 0.16 - 0.8 \cdot 0 = -0.08\,.$$

Now using eq. (5.99) in the book for $k = 0$ we get

$$\gamma(0) - 0.8\gamma(1) = \gamma_{\varepsilon Y}(0) + 0.8\gamma_{\varepsilon Y}(1) - 0.5\gamma_{\varepsilon Y}(2)$$

$$= 0.16 + 0.8 \cdot 0 - 0.5 \cdot (-0.08) = 0.2$$

and for $k = 1$ we get

$$\gamma(1) - 0.8\,\gamma(0) = 0.8\,\gamma_{\varepsilon Y}(0) - 0.5\,\gamma_{\varepsilon Y}(1) = 0.128$$

obtaining

$$\gamma(0) = 0.84,\, \gamma(1) = 0.8\,.$$

For $k = 2$ we get

$$\gamma(2) = 0.8\gamma(1) + \gamma_{\varepsilon Y}(0) = 0.8$$

and then we obtain

$$\gamma(k) = 0.8\gamma(k-1), \quad k > 2\,.$$

As such we get the autocovariance function

$$\gamma(k) = \begin{cases} 0.84 & \text{for } k = 0 \\ 0.8 & \text{for } k = 1{,}2 \\ 0.8\,\gamma(k-1) & \text{for } k > 2 \end{cases} \tag{2}$$

and from that we also obtain the autocorrelation function

$$\rho(k) = \begin{cases} 1 & \text{for } k = 0 \\ \frac{20}{21} & \text{for } k = 1{,}2 \\ 0.8\,\rho(k-1) & \text{for } k > 2 \end{cases} . \tag{3}$$

A plot of the function can be seen on figure 2. For the partial autocorrelation we calculate the following vector[1]:

$$\Phi_k = \Gamma_k^{-1}\delta_k \tag{4}$$

where the Γ_k^{-1} is the k x k autocovariance matrix such that $\Gamma_k = [v_{ij}]$ where $v_{ij} = \gamma(|i-j|)$ and the k x 1 vector δ_k contains the autocovariances from lag 0 to k. The PACF for lag k is then defined as the last element in Φ_k. A plot of the calculated PACF's can be seen on figure 3.

2.1.3

Simulating 10 realizations of our ARMA(1,2) process gives the following result:

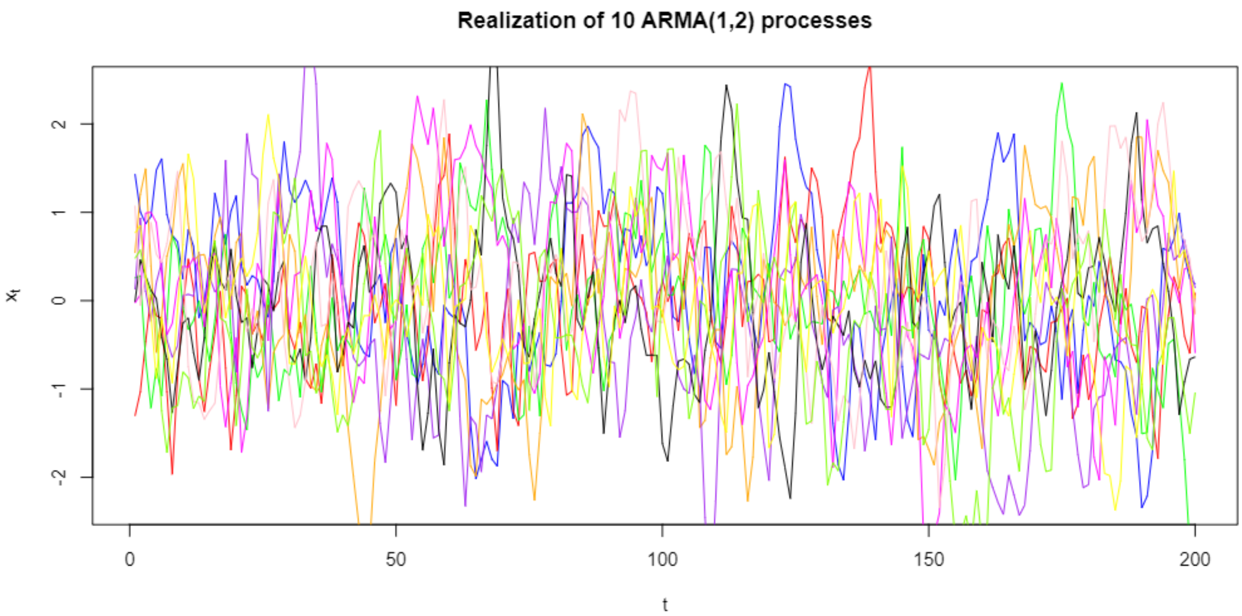

Figure 1: 10 realizations of an ARMA(1,2) process where different colours correspond to the different time series.

We see quite some variation but no obvious global trend in any of the realizations which makes sense since they should come from a stationary process.

[1]This definition of the PACF is taken from https://real-statistics.com/time-series-analysis/stochastic-processes/partial-autocorrelation-function/

2.1.4

The estimated autocorrelation function for each realization can be seen in the following figure:

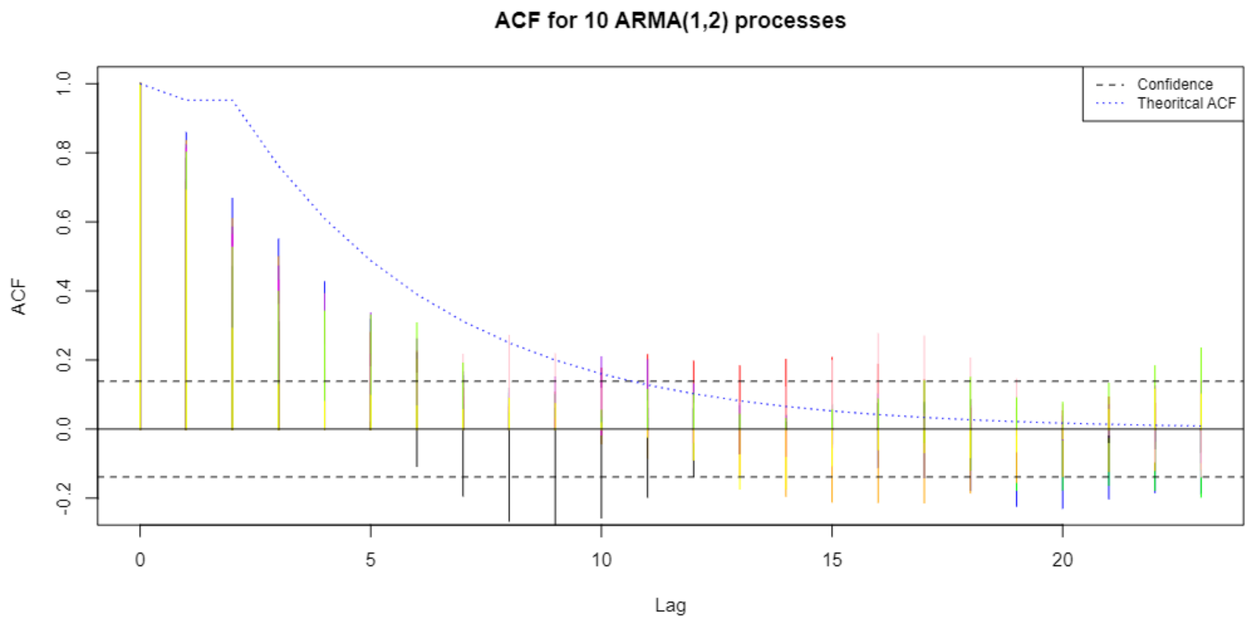

Figure 2: Estimated ACF for each realization using the same colours as in Figure 1. We have also showed the theoretical ACF and and a line showing a 0.05 level of statistical significance (confidence level assuming white noise)

As expected we see a exponential decline as the lag increases with some of the realizations showing damped periodic patterns.

2.1.5

Now we show the estimated partial autocorrelation:

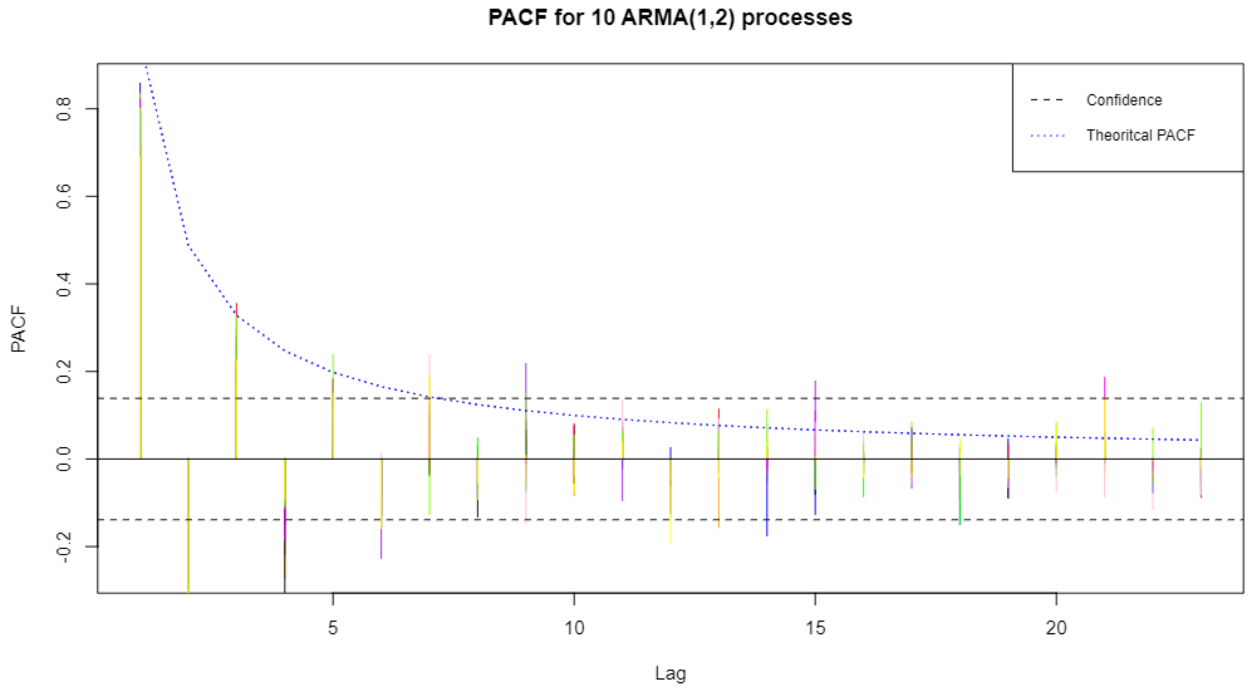

Figure 3: Estimated PACF for each realization using the same colours as in Figure 1. We have also showed the theoretical PACF and and a line showing a 0.05 level of statistical significance (confidence level assuming white noise)

Again we see an exponential decreasing pattern, but with some more clear periodic tendencies compared with the ACF estimate.

2.1.6

The variance for each of the realizations can be seen in the following table:

Realization no.	1	2	3	4	5	6	7	8	9	10
Variance	0.78	0.97	0.70	0.82	0.94	0.79	0.74	0.71	0.95	0.68

Table 1: Variance for 10 realizations of an ARMA(1,2) process

The average variance is 0.81

2.1.7

We generally see some difference in our 10 realizations, but their behaviour are comparable to our analytical calculations. The variances vary quite a bit as seen by Table 1, but the average variance of 0.81 is not far from our analytical result of $\gamma(0) = 0.84$. Looking at Figure 2 we see that for the first 7 lags the analytical ACF is above all the estimated, which makes sense since we are using a non-central estimator which has the expected value $(1 - \frac{|k|}{N})\gamma(k)$ for lag k and N observations. We see that the estimated autocorrelaitons tend to stay inside the 0.05 significance level for large lags, and the ones which are occasionally outside this bound we contribute to random correlations in the estimates. The same comments can be said about the PACF.

Question 2.2

The quarterly number of sales apartments in the capital region has been modelled by.

$$(1 - 1.04B + 0.2B^2)(1 - 0.86B^4)(Y_t - \mu) = (1 - 0.42B^4)\epsilon_t \tag{5}$$

where ϵ_t is white-noise process with variance

$$\sigma_\epsilon^2 = 36963$$

and estimated mean

$$\mu = 2070.$$

We identify that our model can be written in the form of a stationary seasonal model as given on slide 14 from week 5 as

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)\varepsilon_t.$$

From an annual period we get $s = 4$ and we use transformed data with the mean μ subtracted. We have the polynomials

$$\phi(X) = 1 - 1.04X + 0.2X^2, \quad \Phi(X) = 1 - 0.86X$$

and

$$\theta(X) = 1, \quad \Theta(X) = 1 - 0.42X\ .$$

Since

$$\phi(z^{-1})\Phi(z^{-4}) = 0$$

gives the roots

$$z \in \{0.96, 0.96i, -0.96, -0.96i, 0.79, 0.25\}$$

then we have stationarity by slide 13 week 5.

2.2.1

Then we set

$$a = 1.04, \quad b = 0.2, \quad c = 0.86, \quad d = 0.42.$$

and write out equation (5) and isolate Y_t, getting

$$Y_t = aY_{t-1} - bY_{t-2} + cY_{t-4} - acY_{t-5} + bcY_{t-6} + \epsilon_t - d\epsilon_{t-4} \tag{6}$$

Then we assume that

$$Y_t = 0 \quad \text{for } t < 1$$

$$\epsilon_t = 0 \quad \text{for } t < 1$$

We calculate the one-step-predictions for our observations, by ranging t from 0 to 19 in the expression for Y_{t+1}

$$Y_{t+1} = aY_t - bY_{t-1} + cY_{t-3} - acY_{t-4} + bcY_{t-5} + \epsilon_{t+1} - d\epsilon_{t-3}.$$

We then find the expectation of Y_{t+1} given previous observations up to time t. Since $E[\epsilon_{t+1}|Y_t,Y_{t-1},...] = 0$ we have

$$\hat{Y}_{t+1|t} = E[Y_{t+1}|Y_t,Y_{t-1},...] = aY_t - bY_{t-1} + cY_{t-3} - acY_{t-4} + bcY_{t-5} - d\epsilon_{t-3}$$

where we obtain ϵ_{t-3} recursively as

$$\epsilon_{t-3} = Y_{t-3} - aY_{t-4} + bY_{t-5} - cY_{t-7} + acY_{t-8} - bcY_{t-9} + d\epsilon_{t-7}$$

using eq. (6).

The results are shown in Figure 4

When predicting 2019Q1 and 2019Q2 we want to find $\hat{Y}_{21|20}$ and $\hat{Y}_{22|20}$. Were $\hat{Y}_{22|20}$ is calculated as a two-step prediction.

We calculate prediction interval using eq. (5.151) in the book, giving

$$\hat{Y}_{t+k|t} \pm u_{0.025}\sqrt{Var[e_{t+k|t}]}$$
$$= \hat{Y}_{t+k|t} \pm u_{0.025}\sigma_\epsilon\sqrt{1 + \psi_1^2 + ... + \psi_{k-1}^2}$$

where ψ_i are the coefficients in the MA-form

$$Y_t = \epsilon_t + \psi_1\epsilon_{t-1} + \psi_2\epsilon_{t-2} + ...$$

As such for our one-step-predictions the prediction intervals are constant as we get

$$\hat{Y}_{t+1|t} \pm u_{0.025}\sigma_\epsilon.$$

For the two step prediction interval we will first find ψ_1 by rewriting the process to MA-form. In equation (6) we insert the equivalent of Y_{t-1} of equation (6), hence we have

$$Y_{t-1} = aY_{t-2} - bY_{t-3} + cY_{t-5} - acY_{t-6} + bcY_{t-7} + \epsilon_{t-1} - d\epsilon_{t-5}$$

and insert it s.t.

$$\begin{aligned} Y_t =& aY_{t-1} - bY_{t-2} + cY_{t-4} - acY_{t-5} + bcY_{t-6} + \epsilon_t - d\epsilon_{t-4} \\ =& a \cdot (aY_{t-2} - bY_{t-3} + cY_{t-5} - acY_{t-6} + bcY_{t-7} + \epsilon_{t-1} - d\epsilon_{t-5}) \\ & - bY_{t-2} + cY_{t-4} - acY_{t-5} + bcY_{t-6} + \epsilon_t - d\epsilon_{t-4} \\ =& \epsilon_t + a\epsilon_{t-1} + ... \end{aligned}$$

As ϵ_{t-1} is not part of the expressions of Y_{t-i}, $i > 1$ using eq. (6) then replacing more Y_t's using the equation won't change the coefficient ψ_1, hence

$$\psi_1 = a$$

and we then have the two-step prediction interval

$$\hat{Y}_{t+2|t} \pm u_{0.025}\sigma_\epsilon\sqrt{1 + a^2}.$$

Results are shown in table 2

Prediction	Value	Lower 95% PI	Upper 95% PI
$\hat{Y}_{21\|20}$	2314.125	1937.307	2690.943
$\hat{Y}_{22\|20}$	2279.308	1735.644	2822.972

Table 2: Predicted values for 2019Q1 and 2019Q2 including 95% prediction interval

2.2.2

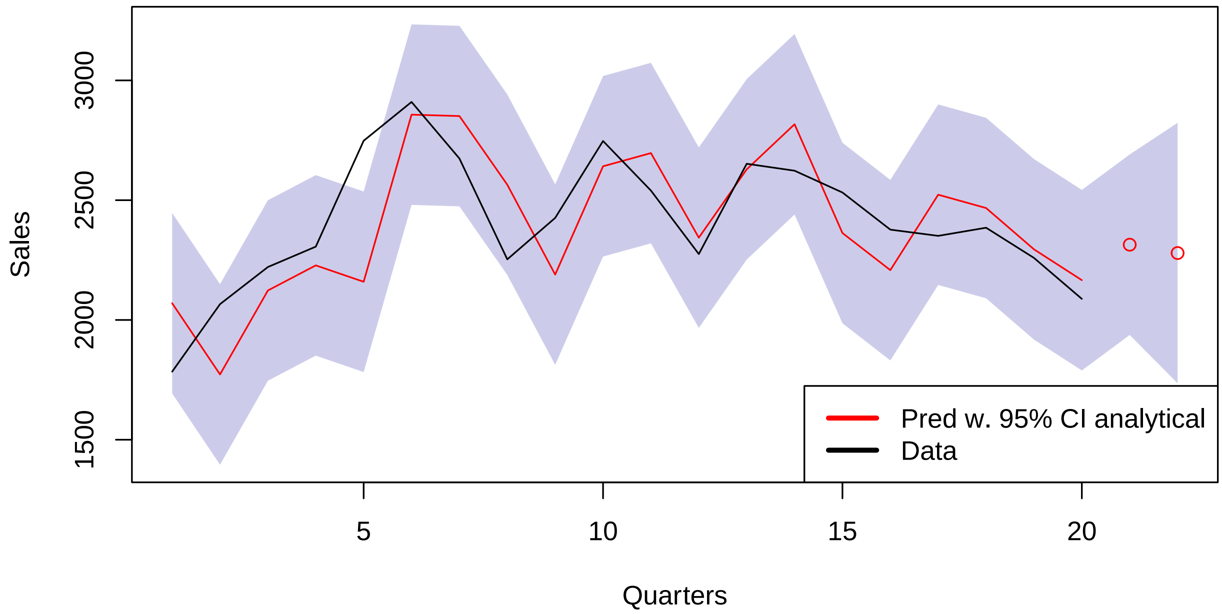

Figure 4: Predictions of quarterly sales along with a 95% prediction interval. Future predictions from table 2 are represented with red circles.

We see that the data fits well with the 95% prediction interval as only one observation lies outside the interval. We notice a trend that the predictions seem to lag one time step behind the data. This trend is especially noticeable for the first half of the predictions. AS expected the prediction interval is larger when predicting two time steps ahead.

Question 2.3

We are given the general ARMA(2,0) process given by:

$$X_t - 1.5X_{t-1} + \phi_2 X_{t-2} = \epsilon_t \tag{7}$$

and we will consider the four variations of this process with different values for ϕ_2 and σ (standard deviation of ϵ_t):

Process	1	2	3	4
ϕ_2	0.52	0.98	0.52	0.98
σ	0.1	0.1	5	5

Table 3: Definition of four variations of the process given by (7)

2.3.1

The process in (7) is given by $\phi(B)X_t = \epsilon_t$ where

$$\phi(B) = 1 - 1.5B + \phi_2 B^2.$$

For $\phi_2 = 0.52$ then $\phi(z^{-1}) = 0$ gives the roots

$$z \in \{0.956, 0.544\}$$

so here the process is stationary. For $\phi_2 = 0.98$ we get the roots

$$z \in \{0.750 + 0.646i, 0.750 - 0.646i\}.$$

and since $\sqrt{0.750^2 + 0.646^2} = 0.990$ the process is also stationary in this case. We see that the process in both cases is invertible since it is an AR process.

2.3.2

For each process defined in Table 3 we simulate 300 observations using the built in R function `arima` which uses conditional-sum-of-squares to find starting values for our parameters and then maximum likelihood. Repeating this 100 times we get the estimates for ϕ_2 as seen in the following histograms:

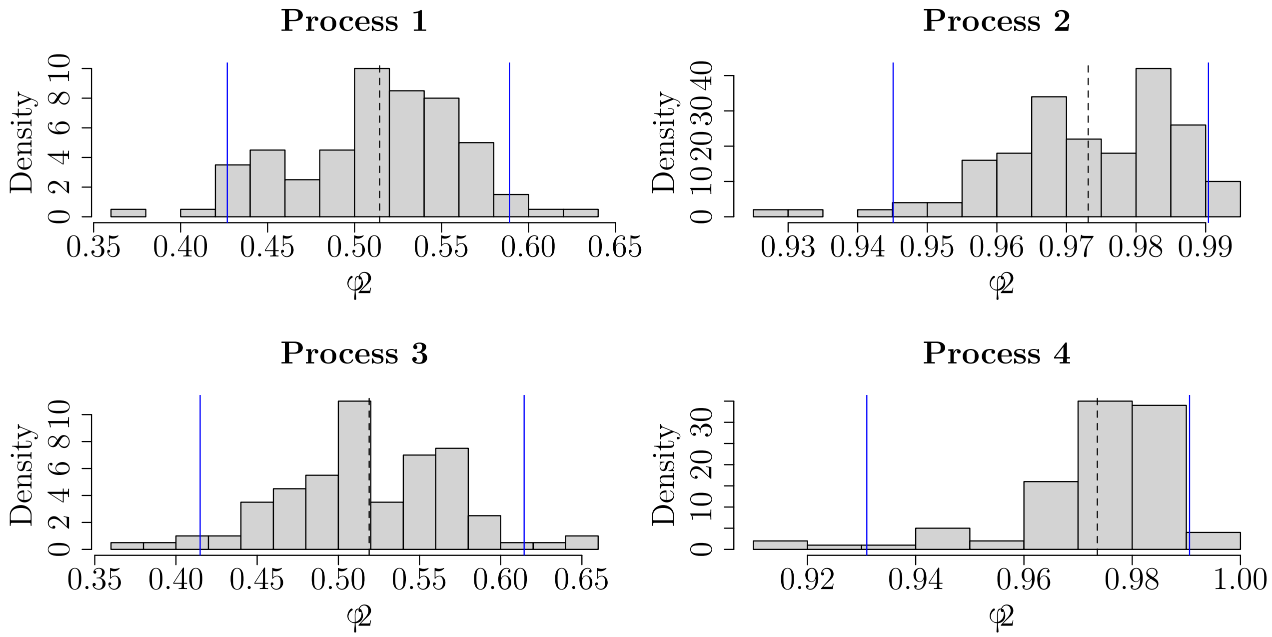

Figure 5: Histograms of the estimates of ϕ_2 for the four processes. The 0.95 confidence interval is indicated with blue lines and the mean value is indicated with the dashed line. The actual parameters (ϕ_2,σ) *are for P1:* (0.52,0.1)*, for P2:* (0.98,0.1)*, for P3:* (0.52,5)*, and for P4:* (0.98,5)*.*

The following table shows the variances, the mean values and quantiles for the estimate of ϕ_2 for the different processes:

Process	1	2	3	4
Variance	2.331e-3	1.637e-4	2.649e-3	2.318e-4
Mean	0.5144	0.9731	0.5190	0.9736
0.025-Quantile	0.4268	0.9451	0.4149	0.9310
0.975-Quantile	0.5890	0.9904	0.6144	0.9906

Table 4: Variance, mean values and quantiles for our estimations of ϕ_2 for the 4 different processes. (ϕ_2,σ) *are for P1:* (0.52,0.1)*, for P2:* (0.98,0.1)*, for P3:* (0.52,5)*, and for P4:* (0.98,5)*.*

2.3.3

From figure 5 and table 4 we can compare Process 1 with 2 and Process 3 with 4 to see the effect of changing ϕ_2 from 0.52 to 0.98. Here we see a more left skewed distribution but a more narrow confidence interval for both cases when $\phi_2 = 0.98$. We similarly see that the variance of the estimates drops for this larger ϕ_2 as can be seen by table 4.

2.3.4

When comparing processes 1 and 3 which both have $\phi_2 = 0.52$ in figure 5 we see that the width is similar but the tails are heavier for process 3 with $\sigma = 5$ and as such the 95 confidence interval is also slightly wider. The variance of the estimations are similar as seen in table 4 but slightly larger for process 3. The differences are not large and it would not be unreasonable if the histograms were generated from the same model.
When comparing processes 2 and 4 that both have $\phi_2 = 0.98$ we notice heavier tails and a wider 95 for process 4 with $\sigma = 5$. This corresponds with the variance being noticeably larger for process 4 as seen in table 4. Still they are very similar and we would argue that they could be generated from the same model.
For both comparisons the variance of the estimations increases for the larger σ value as seen in table 4. Overall we do not see a significant effect on the distribution of estimated ϕ_2 for different values of σ.

2.3.5

The following figure shows scatter plots for our estimations of (ϕ_1,ϕ_2) for each process:

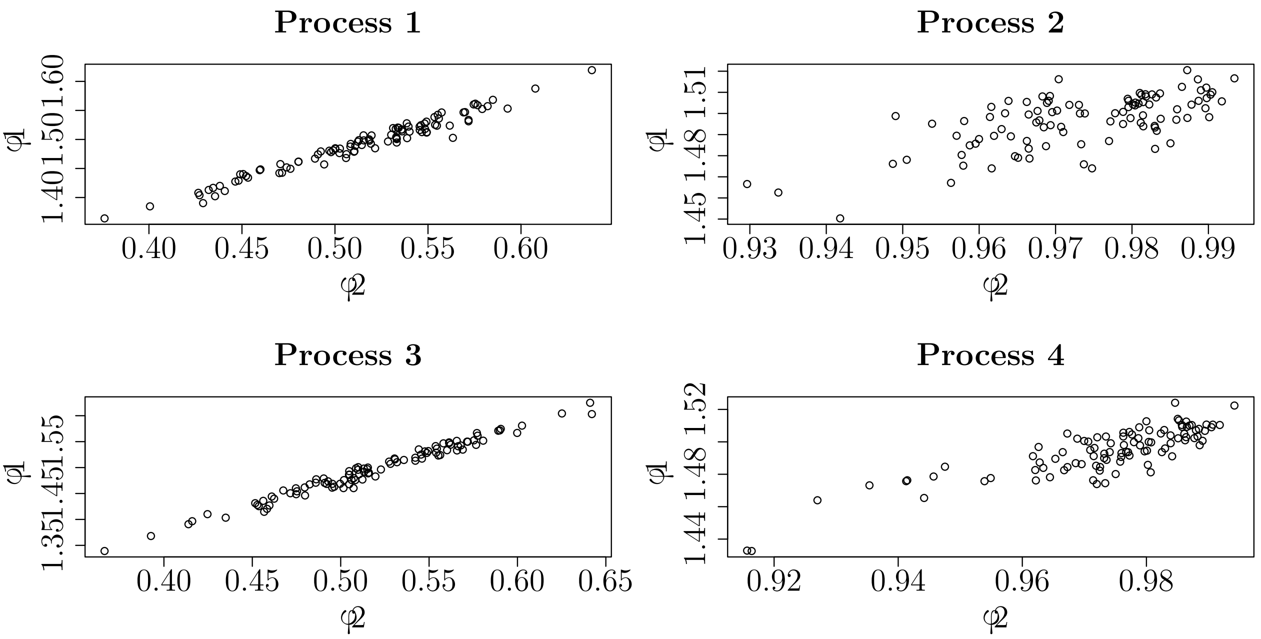

Figure 6: Scatter plots showing our estimations of (ϕ_1,ϕ_2) for the four processes. Again (ϕ_2,σ) are for P1: (0.52,0.1)*, for P2:* (0.98,0.1)*, for P3:* (0.52,5)*, and for P4:* (0.98,5)*.*

We see a noticeably stronger correlation between ϕ_1 and ϕ_2 when $\phi_2 = 0.52$.

2.3.6

As can be seen by our calculation in section 2.3.1 the roots for $\phi(z^{-1})$ lies significantly closer to the unit circle when setting $\phi_2 = 0.98$ compared to 0.52. This means that when simulating with this higher ϕ_2 value the influence from past ϵ_t decreases more slowly compared to using the lower ϕ_2 value. We see that this affects our maximum likelihood estimator with a more skewed distribution of ϕ_2 and a decrease in the correlation between ϕ_1 and ϕ_2 as can be seen from figure 6. The skewness makes sense since if ϕ_2 is just a bit over 0.98 the process becomes not stationary and then it wouldn't be predicted by the maximum likelihood. As mentioned we do not see a huge chance in variance of when changing σ which makes sense since the maximum likelihood estimator should not depend on this. We do however see a noticeably decrease in the variance when setting $\phi_2 = 0.98$ and this could be explained by X_t in (7) having more influence from the past observations and thus the estimator needs fewer observations before it can estimate the parameters.

Appendix: R Code

```
# Q2.1
#simulering
p <- 2 # AR order
q <- 3 # MA order

theta <- c(0.8) #changed from minus because of the way arima.sim works
phi <- c(0.8,-0.5)

cat("AR parameters:",round(theta,3))
cat("MA parameters:",round(phi,3))
n <- 200
colours = list("blue","red","green","orange","purple","black","magenta","pink","
    chartreuse","yellow")
set.seed(1222)
y = matrix(0,10,200)
############################################################
# plotting realizations
for (i in 1:10) {
    # set.seed(i)
    ts <- arima.sim(list(ar=theta,ma=phi), n, sd=0.4)
    y[i,] <- ts[1:200]
    t = c(1:n)
    if (i == 1) {
    plot(t,y[i,],type="l",col=toString(colours[i]),xlab="t",ylab=expression(x[t
        ]),main ="Realization of 10 ARMA(1,2) processes")
    }
    else {
    lines(t,y[i,],type="l",col=toString(colours[i]))
    }
}

###########################################################
#Plotting ACF

for (i in 1:10) {
    ACF <- acf(y[i,],main="ACF",plot = FALSE,type="covariance")
    if (i == 1) {
    plot(ACF$lag,ACF$acf,type="h",col=toString(colours[i]),xlab="Lag",ylab="ACF"
        ,main="ACF for 10 ARMA(1,2) processes")
    }
    else {
    lines(ACF$lag,ACF$acf,type="h",col=toString(colours[i]))
    }

}

#confidence intervals (assuming white noise):
```

```
ci = 0.95
abline(h=qnorm((1 + ci)/2)/sqrt(n),lty=2)
abline(h=-qnorm((1 + ci)/2)/sqrt(n),lty=2)
abline(h = 0)

#theoritical ACF
fun_acf <- function(k) {
    if (k == 0) {
        return(1)
    }
    if (k == 1 | k == 2) {
        return(20/21)
    }
    else {
        return(0.8*fun_acf(k-1))
    }
}

lines(c(0:23),lapply(c(0:23),fun_acf),col="blue",lty=3)
legend("topright",legend=c("Confidence","Theoritcal ACF"),
       col=c("black","blue"),lty = 2:3, cex=0.8)

###############################################################
# PACF
for (i in 1:10) {
    PACF <- pacf(y[i,],main="ACF",plot = FALSE)
    if (i == 1) {
    plot(PACF$lag,PACF$acf,type="h",col=toString(colours[i]),xlab="Lag",ylab="PACF",main="PACF for 10 ARMA(1,2) processes")
    }
    else {
    lines(PACF$lag,PACF$acf,type="h",col=toString(colours[i]))
    }

}
abline(h=qnorm((1 + ci)/2)/sqrt(n),lty=2)
abline(h=-qnorm((1 + ci)/2)/sqrt(n),lty=2)
abline(h = 0)

#Theoritical PACF
fun_pacf <- function(k) {
    rhos <- sapply(c(0:(k-1)),fun_acf)
    rhos <- rhos * 0.84
    Rho_k <- sapply(c(1:k),fun_acf)
    Rho_k <- Rho_k * 0.84
    P_k <- as.matrix(toeplitz(rhos))
    Phi_k <- solve(P_k,Rho_k)
    return(tail(Phi_k,n=1))
}

```

```
lines(c(1:23),lapply(c(1:23),fun_pacf),col="blue",lty=3)
legend("topright",legend=c("Confidence","Theoritcal PACF"),
       col=c("black","blue"),lty = 2:3, cex=0.8)

lapply(c(1:23),fun_pacf)
##################################
#Variance of each realization
vars = rep(0,10)
for (i in 1:10) {
    vars[i] <- var(y[i,])
}
mean(vars)

#################################################
# Q.2.2
library(forecast)
rm(list = ls())
#df <- read.table("A2_sales.txt", header = TRUE)
df <- read.table("Time_Series_2023/Assignment_2/A2_sales.txt", header = TRUE)
plot(df$Sales)
mu <- 2070

ts_sales <- ts((df$Sales-mu), freq = 4) # ts object and transform
phi <- c(1.04,-0.2)
Phi <- c(0.86)
Theta <- c(-0.42)

p <- 2
q <- 0
d <- 0

D <- 0
P <- 1
Q <- 1

Period = 4

model <- arima(ts_sales ,order=c(p,d,q),seasonal = list(order=c(P,D,Q),period=
   Period, fixed=c(phi=phi,
                       Phi=Phi, Theta=Theta)))
pred <- forecast(model,2, level = 95)
plot(df$Sales , xlim = c(1,22), col = "black", type = "l")

points(c(21,22), pred$mean +mu, col = "red")
#plot( c(df$Sales, pred$mean +mu), col = c(rep("black",20),rep("red",2)))

pred$upper # prediction intervals ?
pred$lower

```

```
lines(ts(pred$fitted+mu,freq = 1), col = "blue") # predicted values of Yt

library("plotrix")
plotCI(x = c(21,22),
       y = pred$mean+mu,
       li = pred$lower+mu,
       ui = pred$upper+mu,xlim = c(1,22),ylim = c(1700,3000),col = "red")
lines(df$Sales , col = "black", type = "l")
lines(ts(pred$fitted+mu,freq = 1), col = "blue") # predicted values of Yt
legend("bottomright", legend = c("Prediction with 95% CI","Data","Fitted"),lwd =
    3, col = c("red","black","blue"))







#Samme ting bare med fixed mu.
rm(list = ls())
df <- read.table("A2_sales.txt", header = TRUE)
#df <- read.table("Time_Series_2023/Assignment_2/A2_sales.txt", header = TRUE)
plot(df$Sales)
mu <- 2070

ts_sales <- ts((df$Sales), freq = 4) # ts object and transform
phi <- c(1.04,-0.2)
Phi <- c(0.86)
Theta <- c(-0.42)

p <- 2
q <- 0
d <- 0

D <- 0
P <- 1
Q <- 1

Period = 4


model <- arima(ts_sales ,order=c(p,d,q),seasonal = list(order=c(P,D,Q),period=
   Period,include.mean=TRUE ,fixed=c(phi=phi,

```

```
pred <- forecast(model,2, level = 95)
plot(df$Sales , xlim = c(1,22), col = "black", type = "l")

points(c(21,22), pred$mean, col = "red")
#plot( c(df$Sales, pred$mean +mu), col = c(rep("black",20),rep("red",2)))

pred$upper # prediction intervals ?
pred$lower

lines(ts(pred$fitted,freq = 1), col = "blue") # predicted values of Yt

library("plotrix")
plotCI(x = c(21,22),
       y = pred$mean,
       li = pred$lower,
       ui = pred$upper,xlim = c(1,22),ylim = c(1700,3000),col = "red")
lines(df$Sales , col = "black", type = "l")
lines(ts(pred$fitted,freq = 1), col = "blue") # predicted values of Yt
legend("bottomright", legend = c("Prediction with 95% CI","Data","Fitted"),lwd =
    3, col = c("red","black","blue"))




#Alternativt plot med predictions
library(forecast)
fit <- model
Nile <- c(ts_sales)
upper <- fitted(fit) + 1.96*sqrt(fit$sigma2)
lower <- fitted(fit) - 1.96*sqrt(fit$sigma2)
plot(df$Sales, type="n", ylim=range(lower,upper))
polygon(c(time(Nile),rev(time(Nile))), c(upper,rev(lower)),
        col=rgb(0,0,0.6,0.2), border=FALSE)
#NIlines(Nile)
lines(c(fitted(fit)),col='red')
#out <- (Nile < lower | Nile > upper)
#points(time(Nile)[out], Nile[out], pch=19)
lines(c(1:20),df$Sales)
```

```
#Trying to make predictions analytically
#isolating Y_t and then taking the conditional expectation.
#For 2.2.1 wouldn't we need to calculate the model-predictions of all the
    previous Y_t's before we can predict? (Since the prediction depends on eps_{t
    -4})
#Lad t=1 og t=1..20 være kendt.
Y<-c(0,0,0,0,0,0,0,df$Sales-mu)

phi <- rev(c(1.04,-0.2,0,0.86,-0.8944,0.172))
theta <- c(1,0,0,0,-0.42)

pred_Y <- c(1:21)*0
length(pred_Y)
eps <- c(1:25)*0
k <- 5
n <- 7

for (t in 0:20) {
  eps[t+k] <- Y[t+n]+(-1)*phi%*%Y[(n+t-6):(t-1+n)]+theta[5]*eps[t-4+k]
  pred_Y[t+1] <- phi%*%Y[(n+t-5):(t+n)]+theta[5]*eps[t+k-3]
}


Yp <- df$Sales-mu



#Check pred_Y - IT IS GOOD!!!
(pred_Y[1])
(pred_Y_1 <- 0) #eps[-3]
(pred_Y[2])
(pred_Y_2 <- 1.04*Yp[1]) #eps[-2]
(pred_Y[3])
(pred_Y_3 <- 1.04*Yp[2]-0.2*Yp[1]) #eps[-1]
(pred_Y[4])
(pred_Y_4 <- 1.04*Yp[3]-0.2*Yp[2]) #eps[0]
(pred_Y[5])
(pred_Y_5 <- 1.04*Yp[4]-0.2*Yp[3]+0*Y[2]+0.86*Yp[1] - 0.42*eps_1)
(pred_Y[6])
(pred_Y_6 <- 1.04*Yp[5]-0.2*Yp[4]+0*Y[3]+0.86*Yp[2]-0.8944*Yp[1]- 0.42*eps_2)
(pred_Y[7])
(pred_Y_7 <- 1.04*Yp[6]-0.2*Yp[5]+0*Y[4]+0.86*Yp[3]-0.8944*Yp[2]+0.172*Yp[1]-
    0.42*eps_3)
(pred_Y[16])
(pred_Y_16 <- 1.04*Yp[15]-0.2*Yp[14]+0*Y[13]+0.86*Yp[12]-0.8944*Yp[11]+0.172*Yp
    [10]- 0.42*eps_12)
```

```r
#Check epsilon - IT IS GOOD!!!
(eps[1+k])
(eps_1 <- (df$Sales-mu)[1])
(eps[2+k])
(eps_2<-Yp[2]-1.04*Yp[1])
(eps[3+k])
(eps_3<-Yp[3]-1.04*Yp[2]+0.2*Yp[1])
(eps[4+k])
(eps_4<-Yp[4]-1.04*Yp[3]+0.2*Yp[2]+0*Yp[1])
(eps[5+k])
(eps_5<-Yp[5]-1.04*Yp[4]+0.2*Yp[3]+0*Yp[2]-0.86*Yp[1]-0.42*eps_1)
(eps[6+k])
(eps_6<-Yp[6]-1.04*Yp[5]+0.2*Yp[4]+0*Yp[3]-0.86*Yp[2]+0.8944*Yp[1]-0.42*eps_2)
(eps[7+k])
(eps_7<-Yp[7]-1.04*Yp[6]+0.2*Yp[5]+0*Yp[4]-0.86*Yp[3]+0.8944*Yp[2]-0.172*Yp
    [1]-0.42*eps_3)
(eps[8+k])
(eps_8<-Yp[8]-1.04*Yp[7]+0.2*Yp[6]+0*Yp[5]-0.86*Yp[4]+0.8944*Yp[3]-0.172*Yp
    [2]-0.42*eps_4)
(eps[12+k])
(eps_12<-Yp[12]-1.04*Yp[11]+0.2*Yp[10]+0*Yp[9]-0.86*Yp[8]+0.8944*Yp[7]-0.172*Yp
    [6]-0.42*eps_8)

lines(c(1:21),pred_Y+mu,col="red")

sum((c(pred$fitted)-df$Sales)^2)
sum((pred_Y[1:20]-df$Sales)^2)

###################################################################
# Q 2.3
set.seed(9999)
phi1 <- c(1.5,-0.52)
phi2 <- c(1.5,-0.98)
sd1 <- 0.1
sd2 <- 5
n <- 300
p1 <- matrix(0,100,3)
p2 <- matrix(0,100,3)
p3 <- matrix(0,100,3)
p4 <- matrix(0,100,3)
for (i in 1:100) {
#Simulations
ts1 <- arima.sim(list(ar=phi1,ma=0), n, sd=sd1)
ts2 <- arima.sim(list(ar=phi2,ma=0), n, sd=sd1)
ts3 <- arima.sim(list(ar=phi1,ma=0), n, sd=sd2)
ts4 <- arima.sim(list(ar=phi2,ma=0), n, sd=sd2)
#Estimation of parameters
arima1 <- arima(ts1[1:n],order=c(2,0,0),include.mean=FALSE)
arima2 <- arima(ts2[1:n],order=c(2,0,0),include.mean=FALSE)
arima3 <- arima(ts3[1:n],order=c(2,0,0),include.mean=FALSE)
```

```r
arima4 <- arima(ts4[1:n],order=c(2,0,0),include.mean=FALSE)
# phi_2 values
p1[i,1] <- -arima1$coef[2]
p2[i,1] <- -arima2$coef[2]
p3[i,1] <- -arima3$coef[2]
p4[i,1] <- -arima4$coef[2]
# phi_1 values
p1[i,2] <- -arima1$coef[1]
p2[i,2] <- -arima2$coef[1]
p3[i,2] <- -arima3$coef[1]
p4[i,2] <- -arima4$coef[1]
#variances
p1[i,3] <- arima1$var.coef[2,2]
p2[i,3] <- arima2$var.coef[2,2]
p3[i,3] <- arima3$var.coef[2,2]
p4[i,3] <- arima4$var.coef[2,2]
}

#Histograms
par(mfrow=c(2,2))
hist(p1[,1],main="Process 1", breaks = 10,
xlab=expression(phi[2]),freq=FALSE)
abline(v=quantile(p1[,1],probs=c(0.025,0.975))[1], col="blue")
abline(v=quantile(p1[,1],probs=c(0.025,0.975))[2], col="blue")
abline(v=mean(p1[,1]),col="black",lty=2)

hist(p2[,1],main="Process 2", breaks = 10,
xlab=expression(phi[2]),freq=FALSE)
abline(v=quantile(p2[,1],probs=c(0.025,0.975))[1], col="blue")
abline(v=quantile(p2[,1],probs=c(0.025,0.975))[2], col="blue")
abline(v=mean(p2[,1]),col="black",lty=2)

hist(p3[,1],main="Process 3", breaks = 10,
xlab=expression(phi[2]),freq=FALSE)
abline(v=quantile(p3[,1],probs=c(0.025,0.975))[1], col="blue")
abline(v=quantile(p3[,1],probs=c(0.025,0.975))[2], col="blue")
abline(v=mean(p3[,1]),col="black",lty=2)

hist(p4[,1],main="Process 4", breaks = 10,
xlab=expression(phi[2]),freq=FALSE)
abline(v=quantile(p4[,1],probs=c(0.025,0.975))[1], col="blue")
abline(v=quantile(p4[,1],probs=c(0.025,0.975))[2], col="blue")
abline(v=mean(p4[,1]),col="black",lty=2)

#effects of different phi2 and sigmas:

var(p1[,1])
var(p2[,1])
var(p3[,1])
var(p4[,1])
```

```r

mean(p1[,1])
mean(p2[,1])
mean(p3[,1])
mean(p4[,1])

#pair of estimates
par(mfrow=c(2,2))
plot(p1[,1],-p1[,2],main="Process 1",xlab=expression(phi[2]),ylab=expression(phi
    [1]))
plot(p2[,1],-p2[,2],main="Process 2",xlab=expression(phi[2]),ylab=expression(phi
    [1]))
plot(p3[,1],-p3[,2],main="Process 3",xlab=expression(phi[2]),ylab=expression(phi
    [1]))
plot(p4[,1],-p4[,2],main="Process 4",xlab=expression(phi[2]),ylab=expression(phi
    [1]))
```