Time Series Analysis - Assignment 2

Lauge Thode Hermansen
s204111

Tymoteusz Barcinski
s221937

sarphiv
anonymized

Vincent Olesen
s184017

March 2023

Question 1

Question 1.1

We are given the stochastic process $\{X_t\}$ defined by

$$X_t - 0.8X_{t-1} = \varepsilon_t + 0.8\varepsilon_{t-1} - 0.4\varepsilon_{t-2} \tag{1}$$

where ε_t is a white noise process with standard deviation $\sigma = 0.4$.

Clearly, eq. (1) is an ARMA(1, 2) process of the form

$$\begin{aligned}
\phi(\mathrm{B})Y_t &= \theta(\mathrm{B})\varepsilon_t, \\
\phi(z^{-1}) &= 1 - 0.8z^{-1}, \\
\theta(z^{-1}) &= 1 + 0.8z^{-1} - 0.4z^{-2},
\end{aligned}$$

where B denotes the backward shift operator.

Considering the roots of $\phi(z^{-1})$ in z

$$\phi(z^{-1}) = 0 \implies z = 0.8 < 1,$$

it follows from [2, Theorem 5.12], that eq. (1) is stationary, as the roots are within the unit circle.

Similarly, considering the roots of $\theta(z^{-1})$ in z:

$$\theta(z^{-1}) = 0 \implies z \in \{-1.212, 0.4124\},$$

it follows from [2, Theorem 5.12], that eq. (1) is *not* invertible, as the root -1.212 is outside of the unit circle.

Question 1.2

Firstly, we will derive the mean of the process $\{X_t\}$. Recall that ϵ_t is white noise with mean 0 for all t.

$$\begin{aligned} \mathrm{E}[X_t] &= \mathrm{E}[0.8X_{t-1} + \varepsilon_t + 0.8\varepsilon_{t-1} - 0.5\varepsilon_{t-2}] \\ &= 0.8\,\mathrm{E}[X_{t-1}] \\ &= 0.8\,\mathrm{E}[X_t] && \text{(From stationarity)} \\ \implies \mathrm{E}[X_t] &= 0. \end{aligned}$$

We will now consider the autocovariance $\gamma(k)$ of the process $\{X_t\}$. Let ϕ_k and θ_k respectively denote the coefficients of the polynomials $\phi(z^{-1})$ and $\theta(z^{-1})$. Notably, $\phi_k = 0$ for $k > p = 1$ and $\theta_k = 0$ for $k > q = 2$. Also note that $\theta_0 = \phi_0 = 1$.

From [2, Eqn. 5.97]

$$\begin{cases} \gamma_{\varepsilon X}(0) = \theta_0\sigma^2 \\ \gamma_{\varepsilon X}(1) + \phi_1\gamma_{\varepsilon X}(0) = \theta_1\sigma^2 \\ \gamma_{\varepsilon X}(2) + \phi_1\gamma_{\varepsilon X}(1) = \theta_2\sigma^2 \end{cases} \tag{2}$$

From [2, Eqn. 5.99] and [2, Eqn. 5.101]

$$\begin{cases} \gamma(0) + \phi_1\gamma(1) = \theta_0\gamma_{\varepsilon X}(0) + \theta_1\gamma_{\varepsilon X}(1) + \theta_2\gamma_{\varepsilon X}(2) & (\gamma(-1) = \gamma(1)) \\ \gamma(1) + \phi_1\gamma(0) = \theta_1\gamma_{\varepsilon X}(0) + \theta_2\gamma_{\varepsilon X}(1) & \\ \gamma(2) + \phi_1\gamma(1) = \theta_2\gamma_{\varepsilon X}(0) & \\ \gamma(k) + \phi_1\gamma(k-1) = 0, & k > 2 \end{cases} \tag{3}$$

Solving the linear system of equations given by eq. (2) and eq. (3)

$$\begin{cases} \gamma(0) = \mathrm{Var}[X_t] = 0.8400 \\ \gamma(1) = 0.6720 \\ \gamma(2) = 0.4576 \\ \gamma(k) = -\phi_1\gamma(k-1) = 0.8\gamma(k-1), \quad k > 2 \end{cases}$$

Hence, we have determined $\mathrm{E}[X_t]$, $\mathrm{Var}[X_t]$, and the autocovariance of X_t giving us its second order moment representation.

Question 1.3

In the following figure, we have simulated 10 realization of eq. (1).

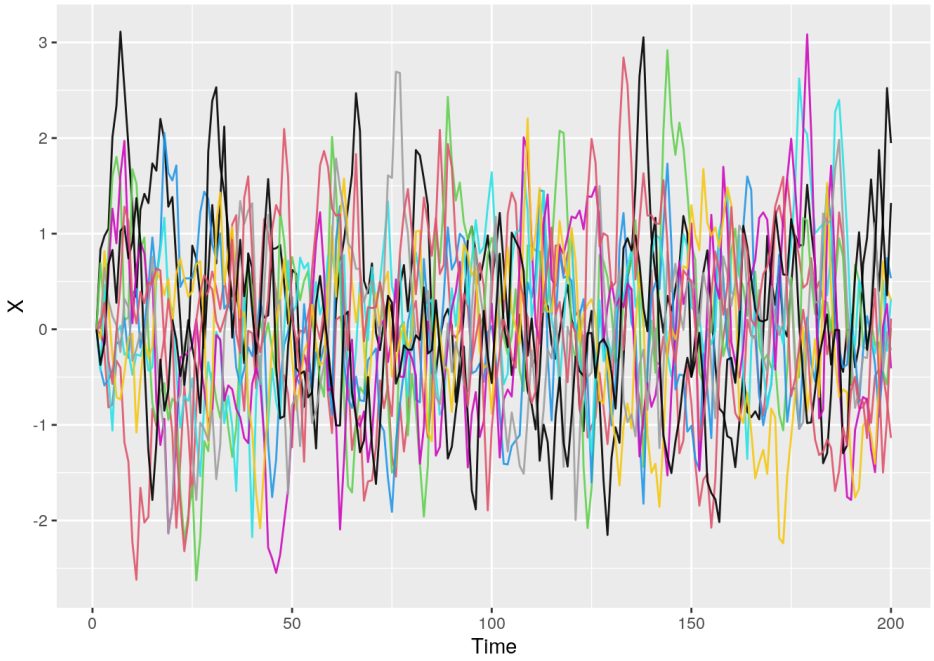

Figure 1: 10 realizations of the ARMA(1, 2) process. Colored by realization.

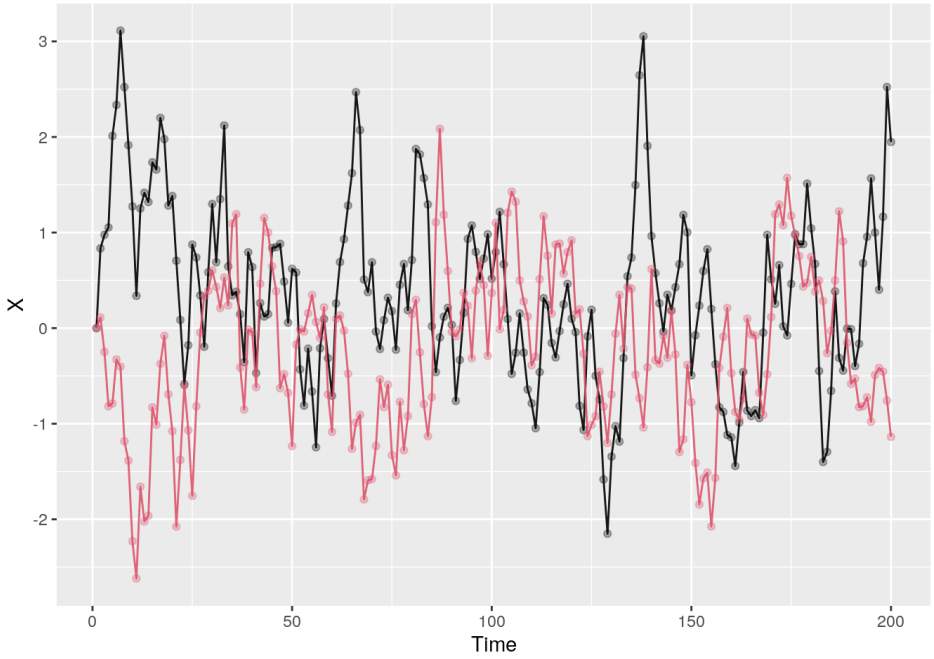

Figure 2: The two first realizations of the ARMA(1, 2) process. Colored by realization.

Questions 1.4

Given the 10 realizations in Question 1.3, we have calculated the empirical auto-correlation function (ACF) for each realization, which are shown in fig. 3.

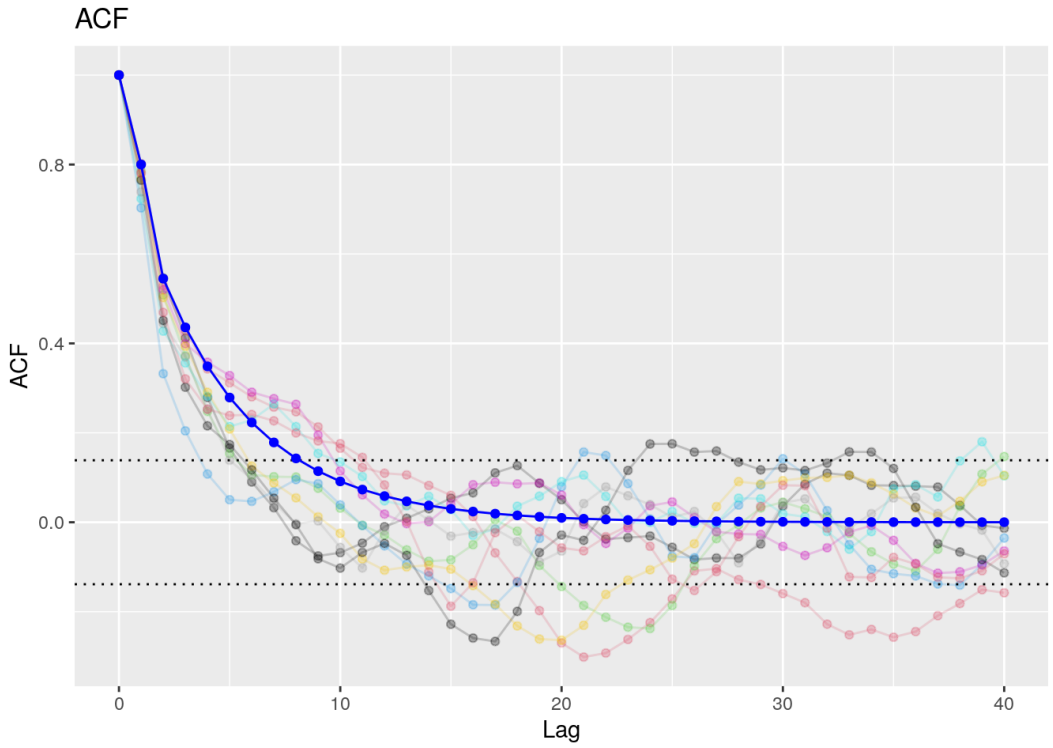

Figure 3: Estimated auto-correlation functions colored by the 10 realizations. The theoretical ACF has been plotted in solid blue. A 95% confidence interval [2, Theorem 6.1] around zero has also been plotted as a dotted black horizontal line.

Question 1.5

Given the 10 realizations in Question 1.3, we have calculated the empirical partial auto-correlation function (PACF) for each realization, which are shown in fig. 4.

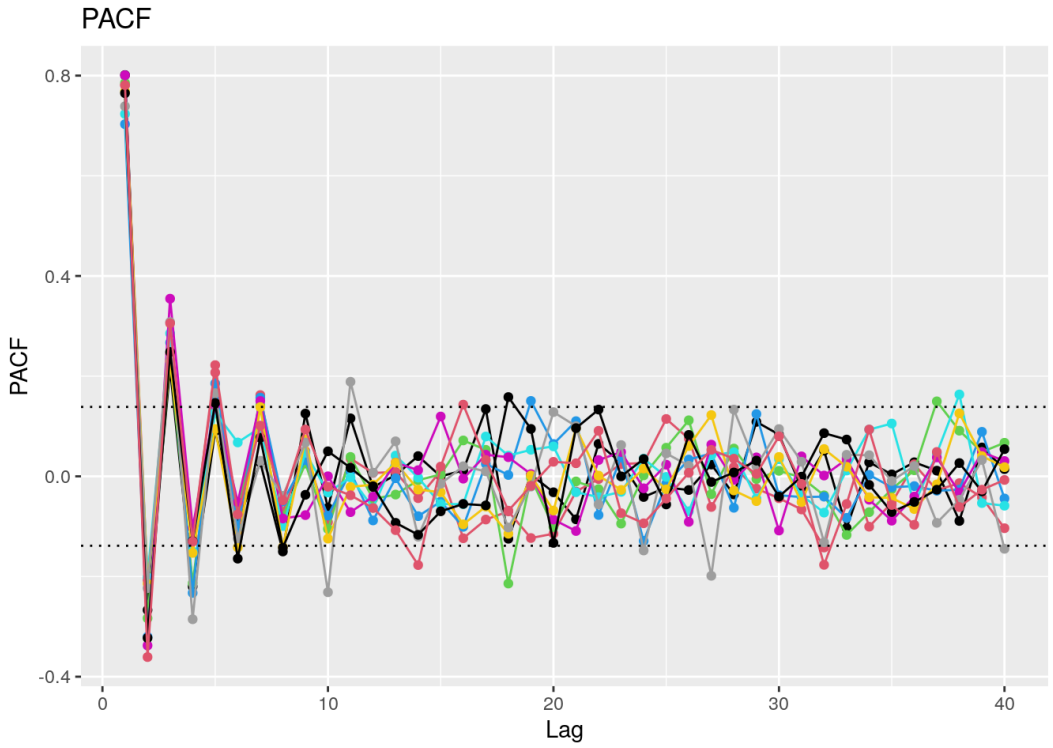

Figure 4: Estimated partial autocorrelation functions colored by the 10 realizations. A 95% confidence interval around zero has also been plotted as a dotted black horizontal line.

Question 1.6

Given the 10 realizations in Question 1.3, we have determined the empirical variance of the process, which is given shown in table 1.

True	Mean	Realizations				
0.8400	0.7753	0.8718	0.7213	0.9574	0.6157	0.5952
		1.0217	0.7180	0.7020	0.7121	0.8384

Table 1: Estimated variances of the 10 realizations

Question 1.7

The theoretical ACF seen on fig. 3 exponentially decreasing approaching 0, which is expected from the positive real root of ϕ [2, Page 122]. This behavior is also evident in the estimated ACFs, where most become insignificant after around 10 lags. The large lag before they become insignificant, indicates that information is propagated forwards for a while, which implies a $\phi(z^{-1})$ root close to length 1. The closer the roots come to having length 1, the closer the process comes to being non-stationary [2, Theorem 5.12] and therefore overall growing/falling away from the mean of 0.

On the PACF plot seen on fig. 4, an oscillation that quickly fades in magnitude to become insignificant can be seen. The strong initial response at $k = 1$ likely stems from the AR(1)

part. Another contributing factor to the initial strong oscillation could be the changing signs from the MA(2) part. It is, however, hard to conclude this from the realizations alone - since the oscillations could have also been caused by complex roots.

The estimated variances of $\{X_t\}$ can be found on table 1, where it can be seen that the variance for each realizations mostly match the theoretical variance. The realizations, however, appear to be slightly below the theoretical, which could simply be due to chance.

Question 2

Questions 2.1, 2.2

We want to predict the quarterly number of sales of apartments in the capital region of Denmark, denoted $\{Y_t\}$, given the quarterly sales from 2014 up to and including 2018. That is $t = 20$ observations.

It is given that

$$\begin{aligned}(1 - 1.04\,\mathrm{B} + 0.2\,\mathrm{B}^2)(1 - 0.86\,\mathrm{B}^4)(Y_t - \mu) \\ = (1 - 0.42\,\mathrm{B}^4)\varepsilon_t\end{aligned} \tag{4}$$

or equivalenty

$$\begin{aligned}(1 - 1.04\,\mathrm{B} + 0.2\,\mathrm{B}^2 - 0.86\,\mathrm{B}^4 + 0.8944\,\mathrm{B}^5 - 0.172\,\mathrm{B}^6)(Y_t - \mu) \\ = (1 - 0.42\,\mathrm{B}^4)\varepsilon_t\end{aligned}$$

where $\mathrm{E}[Y_t] = \mu = 2070$ and $\{\varepsilon_t\}$ is white noise with $\mathrm{E}[\varepsilon_t] = 0$ and $\mathrm{Var}[\varepsilon_t] = \sigma^2 = 36963$. Clearly eq. (4) is a multiplicative $\mathrm{ARMA}(2,0) \times (1,1)_4$ seasonal model. From the roots[1] of ϕ and θ we can see that the process is invertible and stationary.

To ease derivations, we define $W_k = Y_k - \mu$, such that $\mathrm{E}[W_k] = 0$. That is

$$\begin{aligned}W_k &= 1.04W_{k-1} - 0.2W_{k-2} \\ &\quad + 0.86W_{k-4} - 0.8944W_{k-5} + 0.172W_{k-6} \\ &\quad + \varepsilon_k - 0.42\varepsilon_{k-4}\end{aligned} \tag{5}$$

From [2, Theorem 5.15], the optimal prediction is $\hat{W}_{k+1} = \mathrm{E}[W_{k+1} \mid W_k, \ldots, W_1]$.

Further, we have that[2]

$$\begin{aligned}\mathrm{E}[W_k \mid W_t, \ldots, W_1] &= \begin{cases} 0, & k < 1 \\ W_k, & 1 \le k \le t \\ \hat{W}_k, & k > t \end{cases} \\ \mathrm{E}[\varepsilon_k \mid W_t, ..., W_1] &= \begin{cases} 0, & k < 1 \\ W_k - \hat{W}_k, & 1 \le k \le t \\ 0, & k > t \end{cases}.\end{aligned} \tag{6}$$

From eq. (6), we can determine $\mathrm{E}[W_k \mid W_t, \ldots, W_1]$ for $k = 1..t+2$ iteratively. That is, the approach is merely iteratively taking the expectation of eq. (5), see Appendix, as all terms in each step is defined. The results are shown in fig. 5 and the prediction is given in Table 2.

We will now determine the conditional variance of $\hat{W}_{t+1}$ and $\hat{W}_{t+2}$ given W_t, ..., W_1.

[1] $\phi(z^{-1}) = 0 \Rightarrow |z| \in [0.963, 0.963, 0.963, 0.963, 0.785, 0.255]$, and
$\theta(z^{-1}) = 0 \Rightarrow |z| \in [0.805, 0.805, 0.805, 0.805]$.

[2] The first case is a simplifying assumption. As the process is invertible, we can determine a distribution of Y_k for $k < 1$ by construction of a Markov chain. This is outside of the scope of this course, but a canonical reference for the approach is Pinsky and Karlin, 2011.

For $1 \leq k \leq t+1$ we have that

$$\begin{aligned}\text{Var}[\hat{W}_{k+1} \mid W_k, \ldots, W_1] &= \text{Var}[\varepsilon_k \mid W_k, \ldots, W_1] \\ &= \sigma^2\end{aligned}$$

since the remaining terms are constant at time k. This is in accordance with [2, Eqn. 5.150].

For the two-step prediction error, we have that

$$\begin{aligned}\text{Var}[\hat{W}_{t+2} \mid W_t, \ldots, W_1] = \text{Var}[&1.04\hat{W}_{t+1} - 0.2W_t \\ &+ 0.86W_{t-2} - 0.8944W_{t-3} + 0.172W_{t-4} + \\ &\varepsilon_{t+2} - 0.42\varepsilon_{t-2} \mid W_t, \ldots, W_1]\end{aligned}$$

As the variance of the constant terms are zero, it follows that

$$\begin{aligned}\text{Var}[\hat{W}_{t+2} \mid W_t, \ldots, W_1] &= \text{Var}[1.04\hat{W}_{t+1} \mid W_t, \ldots, W_1] + \text{Var}[\varepsilon_{t+2} \mid W_t, \ldots, W_1] \\ &= (1 + 1.04^2)\sigma^2\end{aligned}$$

This is in accordance with [2, Eqn. 5.150], as the first ψ-weight of $\{Y_t\}$ is 1.04, using `ARMAtoMA` in `R`.

Confidence intervals can now be found via [2, Equation 5.151], and the final predictions are found by simply setting $\hat{Y}_k = \hat{W}_k + \mu$.

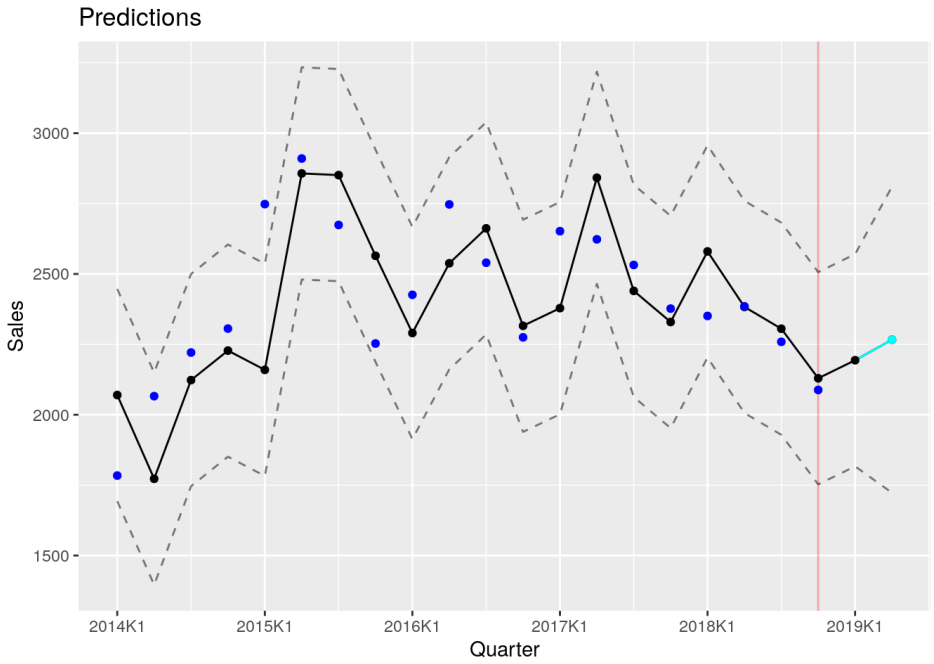

Figure 5: Predictions plotted as black dots with lines between them. The dashed lines are a 95% prediction interval, while the blue dots are the real values of Y_t. The red line marks when test predictions are made. Note that the last, cyan, prediction is a two-step prediction, so the prediction interval is a 2-step interval.

The model appears to predict harmonic and trends in the data well. The confidence intervals are, however, rather large and may suffer from overcoverage. The confidence intervals for the final prediction appears to grow fast because it is a two-step prediction and therefore also incorporates the uncertainty from the previous time step.

	Iterative			Using `R arima`		
t	$\hat{Y}_t$	Lower PI	Upper PI	Y_t	Lower PI	Upper PI
2019Q1	2193.663	1816.845	2570.481	2200.279	1823.414	2577.144
2019Q2	2266.559	1722.895	2810.223	2272.447	1728.758	2816.136

Table 2: One and two step predictions and CI from fig. 5 estimated using the derived approach and using R.

Question 3

We are given a AR(2) process $\{X_t\}$ of the form

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

that is

$$\phi(B)X_t = \varepsilon_t, \quad \phi(B) = 1 - 1.5B + \phi_2 B^2$$

where B denotes the backward shift operator. Further $\phi_2 \in \{0.52, 0.98\}$ and $\sigma^2 = \text{Var}[\varepsilon_t] \in \{0.1^2, 5^2\}$.

Question 3.1

We want to determine stability of eq. (7) given the different values of ϕ_2 and σ. That is we determine the roots of $\phi(z^{-1})$ wrt. z.

$$\phi(z^{-1}) = 1 - 1.5z^{-1} + \phi_2 z^{-2} = 0 \Rightarrow \tag{8}$$

$$z = \frac{3}{4} \pm \frac{\sqrt{9 - 16\phi_2}}{4}. \tag{9}$$

Thus, for $\phi_2 = 0.52$,

$$z \in \{0.544, 0.956\} \Rightarrow |z|1 \in \{0.544, 0.956\} < 1,$$

and for $\phi_2 = 0.98$,

$$z \in \{0.75 - 0.646i, 0.75 + 0.646i\} \Rightarrow |z| = 0.990 < 1.$$

So as we can see, all the roots are within the unit circle, and thus eq. (7) is stationary for both values.

Question 3.2, 3.3, 3.4

We have a collection of stochastic processes $(\{X_t\}_1, \{X_t\}_2, \ldots, \{X_t\}_N)$ which are all AR(2) processes with the same parameters (ϕ_1, ϕ_2, σ). The goal is to estimate those parameters however here we will only focus on the estimation for ϕ_2. The following considerations carry over gracefully to estimation in the multidimensional case when estimation on multiple parameters is conducted. We define an estimator $f(\{X_t\}_i) = \hat{\phi}_{2,i}$ as a maximum likelihood estimator for $i = 1, \ldots, N$. Since we have $N = 100$ stochastic processes available we can define a new estimator f^* in the following way

$$f^*(\{X_t\}_1, \{X_t\}_2, \ldots, \{X_t\}_N) = \frac{1}{N}\sum_{i=1}^{N} f(\{X_t\}_i) \sim \mathcal{N}(\phi_2, \frac{\sigma^2_{\phi_2}}{N}) \tag{10}$$

where for the distribution of f^* we used the Central Limit Theorem, so the result holds asymptotically. This allows us to construct confidence intervals for the final estimate. We now observe 100 realizations of AR(2) process. Figure 6 presents the histograms of estimated

$\hat{\phi_2}$ for each pair of true values (ϕ_2, σ) from 100 simulations. The estimation was performed using the Arima function from R which estimates the parameters by the maximum likelihood estimation. Table presents the results for different variations with uncertainties. For the calculation of confidence intervals, the significance level of $\alpha = 5\%$ was used. We note that for $\phi_2 = 0.98$ the estimated value doesn't belong to the 95% confidence interval. We see that the standard deviation of the distribution of estimates $\sigma_{\hat{\phi_2}}$ depends on the true value of ϕ_2 but doesn't seem to depend on σ. Namely, for the value $\phi_2 = 0.98$ the $\sigma_{\phi_2 \hat{=} 0.98}$ is smaller compared than for the value $\phi_2 = 0.52$. Lack of dependence between $\sigma_{\hat{\phi_2}}$ and σ will be further investigated in the next section.

ϕ_2	σ	$\hat{\phi_2}$	$\hat{\sigma}_{\phi_2}$	Lower CI	Upper CI	5% quantile	95% quantile
0.52	0.1	0.5210	0.0444	0.5123	0.5297	0.4481	0.5857
0.52	5	0.5212	0.0534	0.5107	0.5316	0.4305	0.6000
0.98	0.1	0.9692	0.0144	0.9664	0.9721	0.9432	0.9868
0.98	5	0.9681	0.0168	0.9649	0.9714	0.9332	0.9867

Table 3: Estimates of the ϕ_2 parameter with uncertainties along with quantiles

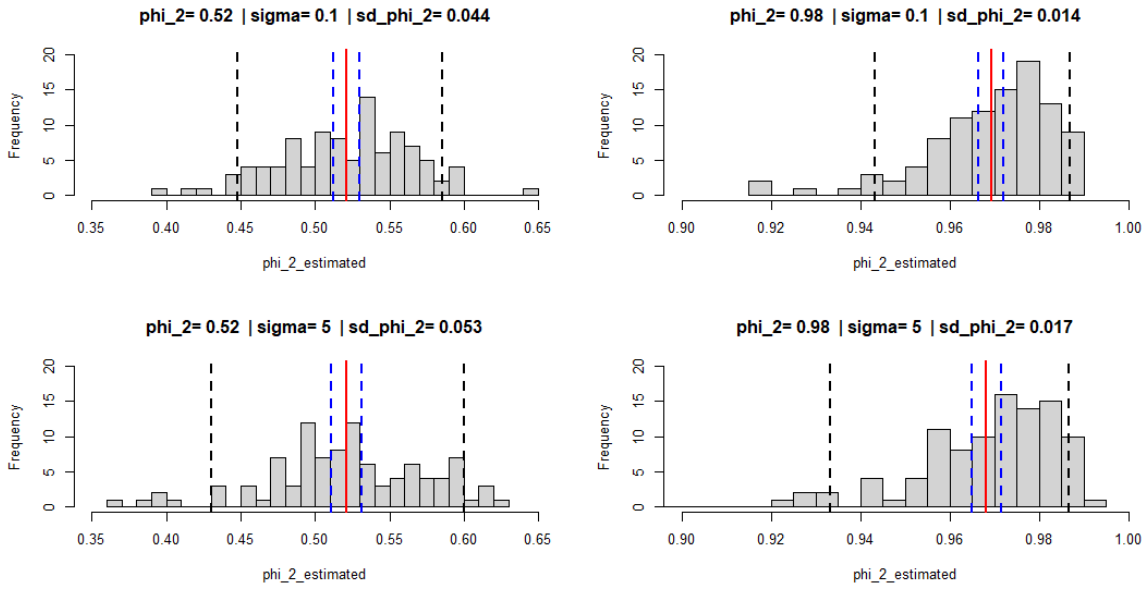

Figure 6: Histograms of estimated $\hat{\phi_2}$ for each pair of true values (ϕ_2, σ) from 100 simulations. The red vertical lines indicates the mean of estimated $\hat{\phi_2}$, the dotted blue lines indicate 95% confidence intervals of the means, and the black dotted lines correspond to 95% quantiles.

Question 3.5

Figure 7 presents the plots of the estimated pairs of parameters (ϕ_1, ϕ_2) for the four variations. We see that for $\phi_2 = 0.52$ the estimates of $(\hat{\phi_1}, \hat{\phi_2})$ are highly corelated. The variance of the distribution of estimates of both parameters doesn't seem to depend on σ.

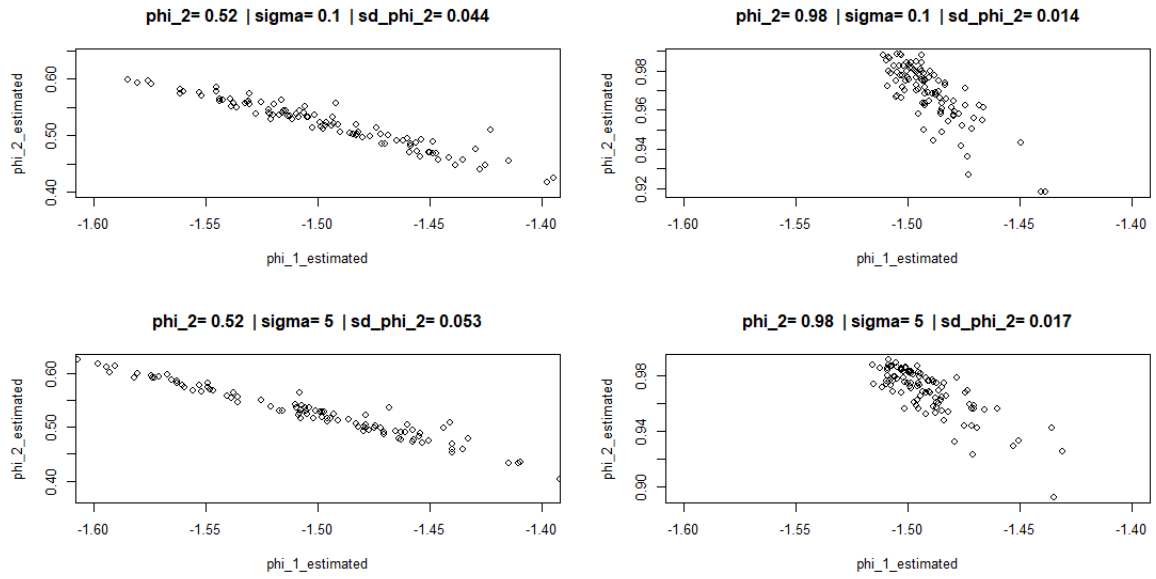

Figure 7: Plots of the estimated pairs of parameters (ϕ_1, ϕ_2) for the four variations.

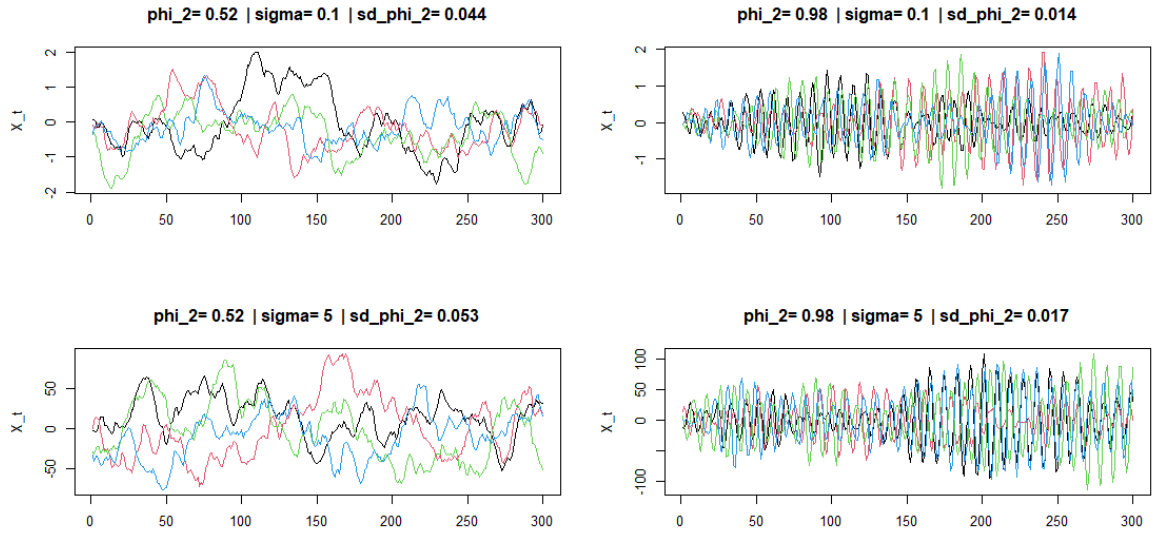

Figure 8: Realizations of the stochastic process X_t for four different variations.

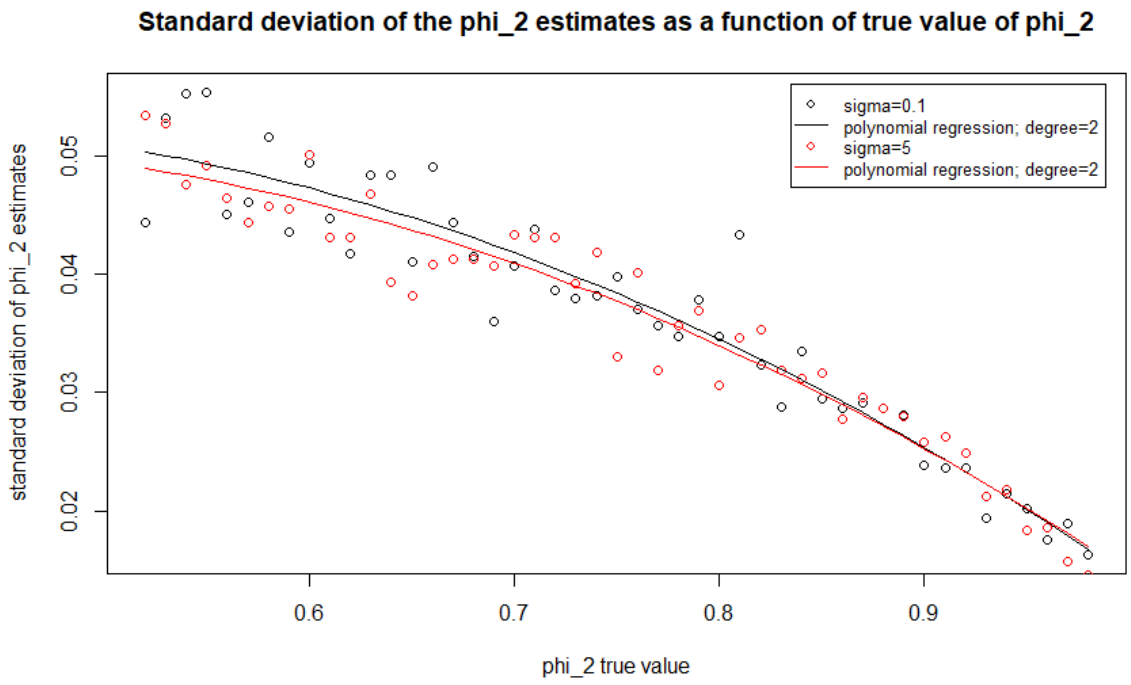

Figure 9: Plot of the standard deviation of the estimated ϕ_2 from 100 simulations as a function of the true value ϕ_2 under different variance values σ of the ϵ_t.

Question 3.6

As we have seen in the previous part of this exercise, different values of σ (residual variance) don't seem to influence the variance of the parameter estimator, but the choice of ϕ_2 influences the variance a lot. Now, we want to elaborate a bit on that.

Influence of σ

We will analyze this theoretically. For that, we use the linear regression formulation of AR in Lasse's slides [1].

The general $AR(p)$ process with residual standard deviation $\sigma > 0$ can be written as

$$Y_t = \sigma\epsilon_t + \sum_{i=1}^{p} \phi_i Y_{t-i}, \quad \epsilon_t \sim \mathcal{N}(0,1) \tag{11}$$

$$\begin{aligned}
\mathbf{Y} &:= (Y_{1+p}, Y_{2+p}, \ldots, Y_T)^\top \\
\boldsymbol{\varepsilon} &:= (\epsilon_{1+p}, \epsilon_{2+p}, \ldots, \epsilon_T)^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\
\mathbf{X} &:= \left[\mathbf{B}^1(\mathbf{Y}), \mathbf{B}^2(\mathbf{Y}), \ldots \mathbf{B}^p(\mathbf{Y})\right] \\
\boldsymbol{\phi} &:= (\phi_1, \phi_2, \ldots \phi_p)^\top
\end{aligned}$$

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\phi} + \boldsymbol{\varepsilon}\sigma.$$

where T be the number of time steps observed of the process, and $\mathbf{B}$ is the usual back-shift operator applied element wise. Notice the slight change in sign convention in eq. (11) compared to the book.

With this formulation, we can solve for the parameters, $\boldsymbol{\phi}$, simply using linear regression, and the variance of the resulting parameter estimator is

$$\text{Var}[\hat{\boldsymbol{\phi}}] = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}. \tag{12}$$

From theorem 5.9 [2] we know that AR processes are always stationary, i.e. eq. (11) can be rephrased as

$$Y_t = \sigma\epsilon_t + \sum_{i=1}^{\infty}\psi_i\epsilon_{t-i}\sigma \tag{13}$$

Now, let $\{\overline{Y_t}\}_{t=1}^T$ be the corresponding stochastic process with residual variance 1 instead of σ. The ψ_i's are still the same, because the parameters in the original process haven't changed. Therefore, using eq. (13) we can find the corresponding linear regression problem for residual variance 1 as follows:

$$\overline{Y_t} = \epsilon_t + \sum_{i=1}^{\infty}\psi_{t-i}\epsilon_{t-i} = \sigma^{-1}Y_t$$
$$\overline{\mathbf{Y}} = \overline{\mathbf{X}}\boldsymbol{\phi} + \boldsymbol{\varepsilon} = \sigma^{-1}\mathbf{X}\boldsymbol{\phi} + \boldsymbol{\varepsilon}$$

Let $\text{Var}[\hat{\overline{\boldsymbol{\phi}}}]$ denote the variance of the parameter estimator in the case of residual variance 1. Using the same formula as in eq. (12), we get

$$\begin{aligned}\text{Var}\left[\hat{\overline{\boldsymbol{\phi}}}\right] &= (\overline{\mathbf{X}}^\top\overline{\mathbf{X}})^{-1} = (\sigma^{-1}\mathbf{X}^\top\sigma^{-1}\mathbf{X})^{-1} = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} \\ &= \text{Var}[\hat{\boldsymbol{\phi}}].\end{aligned}$$

As we see, the variance of the parameter estimator is the same for both variance σ and 1, and since $\sigma > 0$ was chosen arbitrarily it holds for all. Hence, the variance of the parameter estimator is not influenced by the residual variance. It should be noted, however, that this is a biased estimate, but it is consistent, so the conclusions still hold.

Influence of ϕ_2

We will not analyze this theoretically. Instead we made some more simulations to analyze it numerically. We derived (omitted here) that the process is stationary only for $\phi_2 \in]0.5, 1[$, and as seen in the plot 9, the variance of the estimator is much higher for ϕ_2 close to 0.5 than when it's close to 1. We also note, that the relationship between the standard deviation of the distribution of estimates ϕ_2 decreases quadratically as the true value of ϕ_2 increases.

To explain that we turn to fig. 8. We know that both processes are strongly stationary, but as we can see, the fluctuations for $\phi_2 = 0.52$ have longer periods than those for $\phi_2 = 0.98$. This means that for $\phi_2 = 0.52$ we need a longer time horizon (more time steps, i.e. greater T) until the fluctuations even out, than we do for $\phi_2 = 0.98$.

To dive into this theoretically, we would require some frequency analysis. This, however, is at this point in time beyond the scope of the course, so we won't do that.

Let's find the region of ϕ_2 in which the process is stationary, when ϕ_1 is fixed at -1.5. Recall eq. (9) which states that the roots of $\phi(z^-1)$ have the form,

$$z = \frac{3}{4} \pm \frac{\sqrt{9 - 16\phi_2}}{4}.$$

For $9 - 16\phi_2 \geq 0 \Leftrightarrow \phi_2 \leq 9/16$, the roots are real, and thus stationarity implies

$$\left|\frac{3}{4} \pm \frac{\sqrt{9 - 16\phi_2}}{4}\right| < 1 \Rightarrow \frac{3}{4} + \frac{\sqrt{9 - 16\phi_2}}{4} < 1 \Rightarrow 9 - 16\phi_2 < 1 \Rightarrow$$
$$\phi_2 > \frac{1}{2}$$

For $9 - 16\phi_2 < 0 \Leftrightarrow \phi_2 > 9/16$ the roots are complex, so

$$\left|\frac{3}{4} \pm \frac{\sqrt{9 - 16\phi_2}}{4}\right| < 1 \Rightarrow \left|\frac{3}{4}\right|^2 + \left|\frac{\sqrt{9 - 16\phi_2}}{4}\right|^2 < 1 \Rightarrow 9 + 16\phi_2 - 9 < 16 \Rightarrow$$
$$\phi_2 < 1$$

Collecting the two cases, we see that for $\phi_2 \in \ ]\frac{1}{2}, 1[$ the process is starionary. Figure 10 presents the estimated parameters over 100 simulations along with the stable region which was calculated numerically.

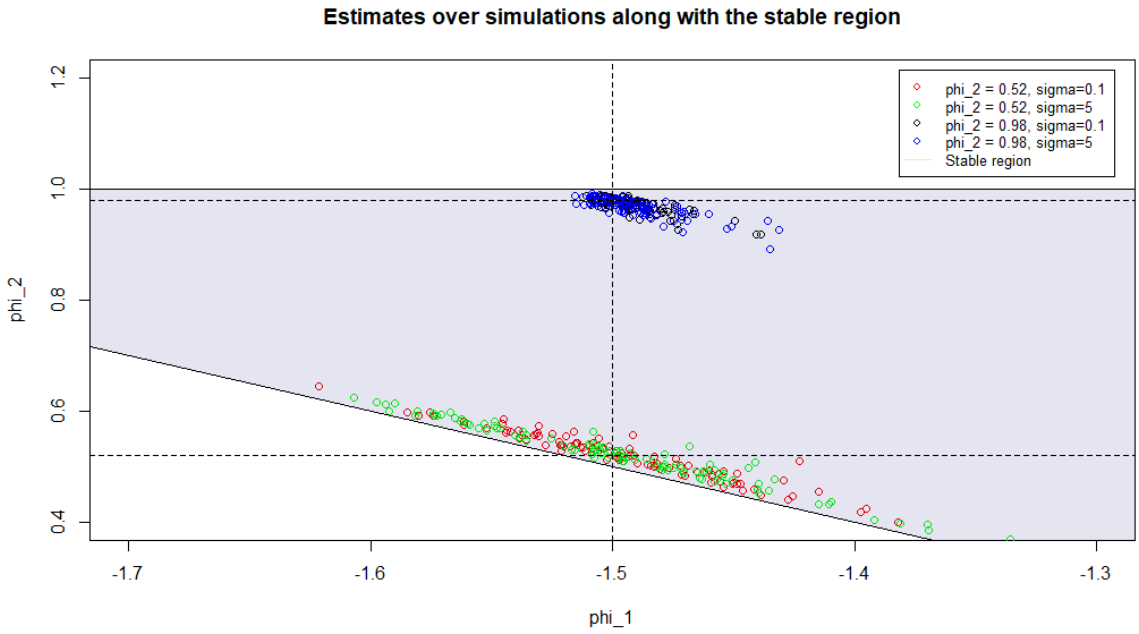

Figure 10: Plot of the estimated (ϕ_1, ϕ_2) over 100 simulations for different configurations of true parameters with the stable region. Stable region means that as long as the pair if (ϕ_2, ϕ_1) is in this region, the process is stationary.

References

[1] Lasse Engbo Christiansenvia. *02417 Lecture 7 part A: Estimating parameters in ARMA models*. 2017. URL: https://youtu.be/IcxMywGiWUc?t=293.

[2] Henrik Madsen. *Time Series Analysis*. Oct. 2008. ISBN: 978-1-4200-5967-0. DOI: 10.1201/9781420059687.

Code

Code A

```
---
title: Assignment 2
author: Author1, Author2, Author3, Author4
output:
  bookdown::html_document2: default
  bookdown::pdf_document2: default
---
```{r include = FALSE}
# Disable inclusion of code chunks in the output
knitr::opts_chunk$set(echo = FALSE)
library("ggplot2")
set.seed(1337)
```


First

Stationary and non-invertible
Roots of $\theta(z^-1)$ are around 0.412 and -1.212. Not
   invertible.
Root of $\phi(z^-1)$ is $4/5$. Stationary.

Two first central moments
Because it is stationary,
the mean and autocovariance function must be invariant to absolute
   time.

\begin{align*}
  E(X_{t}) &=E(0.8X_{t-1} +\epsilon _{t} +0.8\epsilon _{t-1}
     -0.5\epsilon _{t-2})\\
  &=0.8E(X_{t-1})\\
  &=0.8E(X_{t})\\
  &=0
\end{align*}

From (Eq. 5.98)
\begin{align*}
  \gamma _{\epsilon X}(0) &=\theta _{0} \sigma _{\epsilon }^{2}\\
  \gamma _{\epsilon X}(1) &=(\theta _{1} -\phi _{1} \theta _{0})
      \sigma _{\epsilon }^{2}\\
  &\\
```

```
  \gamma _{\epsilon X}( 2) +\phi _{1} \gamma _{\epsilon X}( 1) & =\
     theta _{2} \sigma _{\epsilon }^{2}\\
  \gamma _{\epsilon X}( 2) & =\theta _{2} \sigma _{\epsilon }^{2}
     -\phi _{1}( \theta _{1} -\phi _{1} \theta _{0}) \sigma _{\
     epsilon }^{2}\\
  & =\left( \theta _{2} -\phi _{1} \theta _{1} +\phi _{1}^{2} \
     theta _{0}\right) \sigma _{\epsilon }^{2}
\end{align*}

From (Eq. 5.99), recall that $\displaystyle \gamma ( -1) =\gamma (
   1)$, since it is an even function.
\begin{equation*}
\begin{aligned}
  \gamma ( k) +\phi _{1} \gamma ( k-1) & =\theta _{k} \gamma _{\
     epsilon X}( k) +\theta _{k+1} \gamma _{\epsilon X}( k+1) +\
     theta _{2} \gamma _{\epsilon X}( q-k)\\
  & \\
  \gamma ( 0) +\phi _{1} \gamma ( 1) & =\theta _{0} \gamma _{\
     epsilon X}( 0) +\theta _{1} \gamma _{\epsilon X}( 1) +\theta _
     {2} \gamma _{\epsilon X}( 2)\\
  \gamma ( 1) +\phi _{1} \gamma ( 0) & =\theta _{1} \gamma _{\
     epsilon X}( 0) +\theta _{2} \gamma _{\epsilon X}( 1)\\
  \gamma ( 2) +\phi _{1} \gamma ( 1) & =\theta _{2} \gamma _{\
     epsilon X}( 0)
\end{aligned}
\end{equation*}

From (Eq. 5.101)
\begin{align*}
  \gamma ( 3) +\phi _{1} \gamma ( 2) & =0\\
  \gamma ( k) & =-\phi _{1} \gamma ( k-1) ,\ \forall k\geqslant 3
\end{align*}

Solve for $\displaystyle 0\leqslant k\leqslant 2$ with Maple.
```
```
p1 := -0.8:
q1 := 0.8:
q2 := -0.5:
s := 0.4^2:
ye0 := s:
ye1 := (q1 - p1)*s:
ye2 := (q2 - p1*q1 + p1^2)*s:
solve({
  p1*y1 + y0 = q1*ye1 + q2*ye2 + ye0,
  p1*y0 + y1 = q1*ye0 + q2*ye1,
  p1*y1 + y2 = q2*ye0},
```

```
  {y0, y1, y2}
);
```

Which results in

```
{y0 = 0.8400000000, y1 = 0.6720000000, y2 = 0.4576000000}
```

Simulation 10

```{R}
n_obs <- 200
n_sim <- 10
sigma <- 0.4
phi <- c(-0.8)
p <- length(phi)
theta <- c(0.8, -0.5)
q <- length(theta)
X_sim <- replicate(n_sim, {
  # Create noise
  epsilon <- rnorm(n_obs + q, 0, sigma)
  # Create vector for X
  X <- rep(0, n_obs)
  for (t in seq(2, n_obs)) {
    X[t] <- -phi[1] * X[t - 1] + epsilon[t + q] + theta[1] *
        epsilon[t + q - 1] + theta[2] * epsilon[t + q - 2]
  }
  # Return X
  X
})
# Plot all simulations
# plot.ts(X_sim)
# Plot all simulations
sim_plot <- ggplot(mapping = aes(x = seq(1, n_obs))) +
  labs(x = "Time", y = "X")
for (i in seq(1, n_sim)) {
  sim_plot <- local({
    j <- i
    sim_plot +
      geom_line(aes(y = X_sim[, j]), stat = "identity", alpha =
          0.9, col = j)
      # + geom_point(aes(y = X_sim[, j]), stat = "identity", alpha
          = 0.3, col = j)
  })
}
```

```
sim_plot
```

```{R}
sim_plot <- ggplot(mapping = aes(x = seq(1, n_obs))) +
  labs(x = "Time", y = "X")
for (i in seq(1, 2)) {
  sim_plot <- local({
    j <- i
    sim_plot +
      geom_line(aes(y = X_sim[, j]), stat = "identity", alpha =
          0.9, col = j) +
      geom_point(aes(y = X_sim[, j]), stat = "identity", alpha =
          0.3, col = j)
  })
}
sim_plot
```

ACF
```{R}
n_lag <- 40
# Empirical ACF
acf_matrix <- matrix(0, nrow = n_lag+1, ncol = n_sim)
for (i in seq(1, n_sim)) {
  acf_matrix[, i] <- acf(X_sim[, i], lag.max = n_lag, type = c("
      correlation"), plot = FALSE)$acf
}
# Theoretical ACF
acf_theo <- rep(0, n_lag + 1)
acf_theo[1] <- 0.84
acf_theo[2] <- 0.672
acf_theo[3] <- 0.4576
for (k in seq(3, n_lag)) {
  acf_theo[k + 1] <- -phi[1] * acf_theo[k]
}
acf_theo <- acf_theo / acf_theo[1]
# Plot
# NOTE: Significance level from
acf_sig_level <- qnorm((1 + 0.95) / 2) / sqrt(sum(!is.na(X_sim[,
    1])))
acf_plot <- ggplot(mapping = aes(x = seq(0, n_lag)))
for (i in seq(1, n_sim)) {
  acf_plot <- local({
    j <- i
    acf_plot +
```

```
      geom_line(aes(y = acf_matrix[, j]), stat = "identity", alpha
         = 0.2, col = j) +
      geom_point(aes(y = acf_matrix[, j]), stat = "identity", alpha
          = 0.3, col = j)
  })
}
acf_plot <- acf_plot +
  geom_hline(yintercept = acf_sig_level, linetype = "dotted") +
  geom_hline(yintercept = -acf_sig_level, linetype = "dotted") +
  geom_line(aes(y = acf_theo), stat = "identity", color = "blue") +
  geom_point(aes(y = acf_theo), stat = "identity", color = "blue")
     +
  labs(title = "ACF", x = "Lag", y = "ACF")
acf_plot
```

PACF

```{R}
pacf_matrix <- matrix(0, nrow = n_lag, ncol = n_sim)
for (i in seq(1, n_sim)) {
  pacf_matrix[, i] <- pacf(X_sim[, i], lag.max = n_lag, type = c("
     correlation"), plot = FALSE)$acf
}
pacf_plot <- ggplot(mapping = aes(x = seq(1, n_lag)))
for (i in seq(1, n_sim)) {
  pacf_plot <- local({
    j <- i
    pacf_plot +
      geom_line(aes(y = pacf_matrix[, j]), stat = "identity", col =
          j) +
      geom_point(aes(y = pacf_matrix[, j]), stat = "identity", col
         = j)
  })
}
pacf_plot <- pacf_plot +
  geom_hline(yintercept = acf_sig_level, linetype = "dotted") +
  geom_hline(yintercept = -acf_sig_level, linetype = "dotted") +
  labs(title = "PACF", x = "Lag", y = "PACF")
pacf_plot
```

Variance

```{R}
X_sim_var <- apply(X_sim, 2, var)
# var_plot <- ggplot(mapping = aes(x = seq(1, n_sim), y = X_sim_var
   )) +
```

```
#    geom_bar(stat = "identity", fill = 1:n_sim) +
```

Code B

```
rm(list=ls())
print(utils::getSrcDirectory(function(){}))
print(utils::getSrcFilename(function(){}, full.names = TRUE))
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
# options(scipen=999)
options(scipen=0)
# dev.off()
Sys.setenv(LANG = "en")

library(MASS)
library(dplyr)
# library(tsibble)
library(forecast)
# library(matlib)
library(nlme)

######## 1. STABILITY ##################
n = 200
parameters_1 = c("phi_1" = 0.8, "theta_1" = 0.8, "theta_2" = -0.5,
   "std_noise" = 0.4)
simulations_number = 10
results = matrix(NA, nrow = n, ncol = simulations_number)
for (i in 1:simulations_number){
  results[,i] = arima.sim(n = n, list(ar = parameters_1[1], ma =
     parameters_1[2:3]),
                          sd = parameters_1[4])
}
average_model <- apply(results, MARGIN = 1, mean)
acf_plot <- acf(average_model)
plot(acf_plot)

ARMAacf(ar = parameters_1[1], ma =parameters_1[2:3] , lag.max = 10)

MAX_LAG = 20
par(mfrow=c(1,1))
matplot(results, type = "l", col = "grey")
lines(average_model, col = "red", lwd = 2)

acf_average <- acf(average_model, na.actioin = na.pass, plot=FALSE,
                   lag.max = MAX_LAG)
pacf_average <- pacf(average_model, na.actioin = na.pass, plot=
```

```
    FALSE,
                              lag.max = MAX_LAG)

acf_full <- acf(results, na.action=na.pass, plot=FALSE, lag.max =
    MAX_LAG)
acf_matrix <- t(apply(acf_full$acf, MARGIN = 1, diag))

pacf_full <- pacf(results, na.action=na.pass, plot=FALSE, lag.max =
     MAX_LAG)
pacf_matrix <- t(apply(pacf_full$acf, MARGIN = 1, diag))

par(mfrow=c(1,2))
value = 0.15
matplot(acf_matrix, type = "l", col = "grey")
abline(h = 0, col = "black")
abline(h = value, col = "blue", lty = 2)
abline(h = -value, col = "blue", lty = 2)
lines(acf_average$acf, col = "red", lwd = 2)

value = 0.15
matplot(pacf_matrix, type = "l", col = "grey")
abline(h = 0, col = "black")
abline(h = value, col = "blue", lty = 2)
abline(h = -value, col = "blue", lty = 2)
lines(pacf_average$acf, col = "red", lwd = 2)

autoplot(fit) # fits the roots in the complex unit
# https://otexts.com/fpp2/arima-r.html

######## 2. APARTMENTS ################
df <- read.table("A2_sales.txt", sep="",header=TRUE)
df <- df %>%
  mutate(year = as.numeric(substr(Quarter, 1, 4)),
         quarter = as.numeric(substr(Quarter, 6, 6)),
         index = seq(1, dim(df)[1])) %>%
  mutate(time = as.numeric(year + (quarter - 1)/4))
mu = 2070
period = 4
noise_variance = 36963

par(mfrow=c(1,1))
plot(df$index, df$Sales - mu)


parameters_2 = c("phi_1" = 1.04, "phi_2" = -0.2, "Phi_1" = 0.86,
                 "Theta_1" = -0.42, "mu" = mu)
```

```
model <- arima(df$Sales, order = c(2, 0, 0),
               seasonal = list(order = c(1, 0, 1), period = period)
                   ,
               include.mean = TRUE,
               fixed = parameters_2)
predictions <- predict(model, n.ahead = 2, se.fit = TRUE)

parameters_sarima <- c()
ARMAtoMA(model)

model$sigma2

model_our_sigma <- model
model_our_sigma$sigma2 <- noise_variance
str(model_our_sigma)

predictions_our_sigma <- predict(model_our_sigma, n.ahead = 2, se.
    fit = TRUE)

stuff <- as.numeric(predictions_our_sigma$pred)
std <- as.numeric(predictions_our_sigma$se)
c(stuff[1] - qnorm(0.975)*std[1], stuff[1] + qnorm(0.975)*std[1])
c(stuff[2] - qnorm(0.975)*std[2], stuff[2] + qnorm(0.975)*std[2])


# ? how to set sigma in the arima function?

# using forecast package
df$Sales %>%
  Arima(order = c(2, 0, 0),
        seasonal = list(order = c(1, 0, 1), period = period),
        include.mean = TRUE,
        fixed = parameters_2) %>%
  forecast(h=2) %>%
  autoplot

######## 3. ARMA SIMULATIONS ########################################3
set.seed(400)

n_3 = 300
simulations_number_3 = 100
# phi_2_list = -c(seq(0.52, 0.98, 0.1), 0.98)
# sigma_list =c(0.1, 1, 3, 5)

phi_2_list = c(-0.52, -0.98)
# phi_2_list = -seq(0.52, 0.98, 0.01)
```

```
sigma_list = c(0.1, 5)

parameters_3 = c("phi_1" = 1.5, "phi_2" = NA)
results_3 <- array(NA, dim=c(length(phi_2_list), length(sigma_list)
    ,
                                 n_3, simulations_number_3))
# dim(results_3): (2, 2, 300, 100)
# phi, sigma, n_3, simulations_number_3

# simulations
for(i in 1:length(phi_2_list)){
  for(j in 1:length(sigma_list)){
    results_tmp = matrix(NA, nrow = n_3, ncol = simulations_number_
       3)
    phi_2 = phi_2_list[i]
    sigma = sigma_list[j]
    for (index in 1:simulations_number_3){
      results_tmp[,index] = arima.sim(n = n_3,
                                      list(ar = c(parameters_3[1], phi_
                                         2)),
                                      sd = sigma, n.start=2)
    }
    results_3[i, j, , ] = as.array(results_tmp)
  }
}

estimation_full <- array(NA, dim=c(length(phi_2_list), length(sigma
    _list), 2, simulations_number_3),
                         dimnames = list(phi_2_list, sigma_list,
                                         c("phi_1_estimated", "phi_
                                            2_estimated")))
# phi, sigma, estimation_parameters, simulations_number_3

for(i in 1:length(phi_2_list)){
  for(j in 1:length(sigma_list)){
    print(c(phi_2_list[i], sigma_list[j]))
    for(simulation_number in 1:simulations_number_3){
      series = results_3[i,j, ,simulation_number]
      model_tmp_full <- arima(series, order = c(2, 0, 0),
                              include.mean = FALSE)
      estimation_full[i,j, ,simulation_number] =  model_tmp_full$
         coef[c("ar1", "ar2")]
    }
  }
}
```

```
estimation_full = -estimation_full

par(mfrow = c(1, 1))
plot(estimation_full[1, 1,1,], estimation_full[1, 1,2,], col = "red
    ",
      xlim = c(-1.7, -1.3), ylim = c(0.4, 1.2),
      xlab="phi_1", ylab = "phi_2",
      main = "Estimates over simulations along with the stable
          region")
lines(estimation_full[1, 2,1, ], estimation_full[1, 2,2, ], type =
    "p", col = "green")
lines(estimation_full[2, 1,1, ], estimation_full[2, 1,2, ], type =
    "p", col = "black")
lines(estimation_full[2, 2,1, ], estimation_full[2, 2,2, ], type =
    "p", col = "blue")
polygon(x = c(-2, 2, 0),                                  # X-Coordinates
    of polygon
        y = c(1, 1, -1),                                    # Y-
            Coordinates of polygon
        col = rgb(0, 0, 0.5, alpha = 0.1))
abline(v = -1.5, lty = 2)
abline(h = -phi_2_list,lty = 2)
legend("topright", 95,
       legend=c("phi_2 = 0.52, sigma=0.1",
                "phi_2 = 0.52, sigma=5",
                "phi_2 = 0.98, sigma=0.1",
                "phi_2 = 0.98, sigma=5",
                "Stable region"),
       col=c("red", "green", "black", "blue", rgb(0, 0, 0.5, alpha
           = 0.1)),
       lty = c(NA,NA,NA,NA, 1), cex=0.8,
       inset = 0.02, pch=c(1,1,1,1, NA))

results <- matrix(NA, nrow = 4, ncol = 5)
par(mfrow=c(length(sigma_list), length(phi_2_list)))

phi_2_small_xlim = c(0.35, 0.65)
phi_2_large_xlim =  c(0.9, 1)

i = 1
for(sigma_2_index in 1:length(sigma_list)){
  for(phi_2_index in 1:length(phi_2_list)){

    sd_res <- sd(results_3[phi_2_index, sigma_2_index, , ])
    ifelse(phi_2_list[phi_2_index] == -0.52,
           hist(estimation_full[phi_2_index, sigma_2_index, 2, ],
```

```
            breaks = 25, xlab="phi_2_estimated",
              main = paste("phi_2=", phi_2_list[phi_2_index],
                           "  |  sigma=", sigma_list[sigma_2_index
                             ],
                           "  |  sd_phi_2=", round(sd(estimation_
                             full[phi_2_index, sigma_2_index, 2,
                              ]), 3)),

              ylim = c(0, 20), xlim = phi_2_small_xlim),
          hist(estimation_full[phi_2_index, sigma_2_index, 2, ],
            breaks = 25, xlab="phi_2_estimated",
              main = paste("phi_2=", phi_2_list[phi_2_index],
                           "  |  sigma=", sigma_list[sigma_2_index
                             ],
                           "  |  sd_phi_2=", round(sd(estimation_
                             full[phi_2_index, sigma_2_index, 2,
                              ]), 3)),

              ylim = c(0, 20), xlim = phi_2_large_xlim)
          )

  phi_2_estimated_mean = mean(estimation_full[phi_2_index, sigma_
    2_index, 2, ])
  phi_2_estimated_sd = sd(estimation_full[phi_2_index, sigma_2_
    index, 2, ])
  CI_lower = phi_2_estimated_mean - qnorm(0.975)*phi_2_estimated_
    sd/sqrt(simulations_number_3)
  CI_upper = phi_2_estimated_mean + qnorm(0.975)*phi_2_estimated_
    sd/sqrt(simulations_number_3)
  quantile_empricial = quantile(estimation_full[phi_2_index,
    sigma_2_index, 2, ], 0.95)

  abline(v = phi_2_estimated_mean, col = "red", lty = 1, lwd = 2)
  abline(v = CI_lower, lwd = 2, lty = 2, col = "blue")
  abline(v = CI_upper, lwd = 2, lty = 2, col = "blue")
  abline(v = quantile_empricial, lwd = 2, lty = 2, col = "black")

  results[i, ] <- c(phi_2_estimated_mean, phi_2_estimated_sd, CI_
    lower, CI_upper, quantile_empricial)
  i = i + 1
}
```