Maximum Likelihood Estimation of an AR(1) Process

1st Mandatory Assignment

Advanced Time Series Analysis

December 1, 2019

Jolanda Dünner

5394979

M.Sc. Data Science in Business and Economics

Robert Neulen

4005894

M.Sc. Data Science in Business and Economics

Contents

List of Figures

1 Introduction

There are several ways how parameters of ARMA processes can be estimated. Intuitively, they are expected to yield similar results. In this short paper, the focus is on the estimation of a simple AR(1) process and the different optimization algorithms. We start by simulating an AR(1) process in the first section. In the second part, we analyze and compare different optimization algorithms that are available in MATLAB for the conditional maximum likelihood. Finally, we conduct statistical inference for the estimation setting and give a short conclusion.

2 Model description

In this section, we will generate and discuss the AR(1) process that will be used throughout the paper. Also, we prepare the conditional Log Likelihood function used for the estimation.

2.1 AR(1) Process

We assume that the order of the process is known and defined as an AR process of order 1. The goal is to identify the parameters (c, ϕ, σ^2) that generated the observed time series y_t. As stated by Hamilton (1994, p.117), the MLE approach requires a specific distribution of the innovation term ϵ_t. Knowing the correct joint density function is crucial for maximum likelihood estimation. In this model, the ϵ terms are generated as a Gaussian White Noise, making them stationary and ergodic by definition. Since it is an AR process of order 1, it can be easily confirmed that y_t is a stationary process if $|\phi|$ is smaller than 1. To generate our y_t, we use the process specified in equation (1) with $c = 5$ and $\phi = 0.7$, making them the true values that we try to estimate with the maximum likelihood approach:

$$y_t = 5 + 0.7y_t - \epsilon_t \tag{1}$$

We will use $T = 100$ (length of the time series) and $y_0 = 17$ as the starting point of the time series. The resulting time series with the expected value can be seen in figure 1.

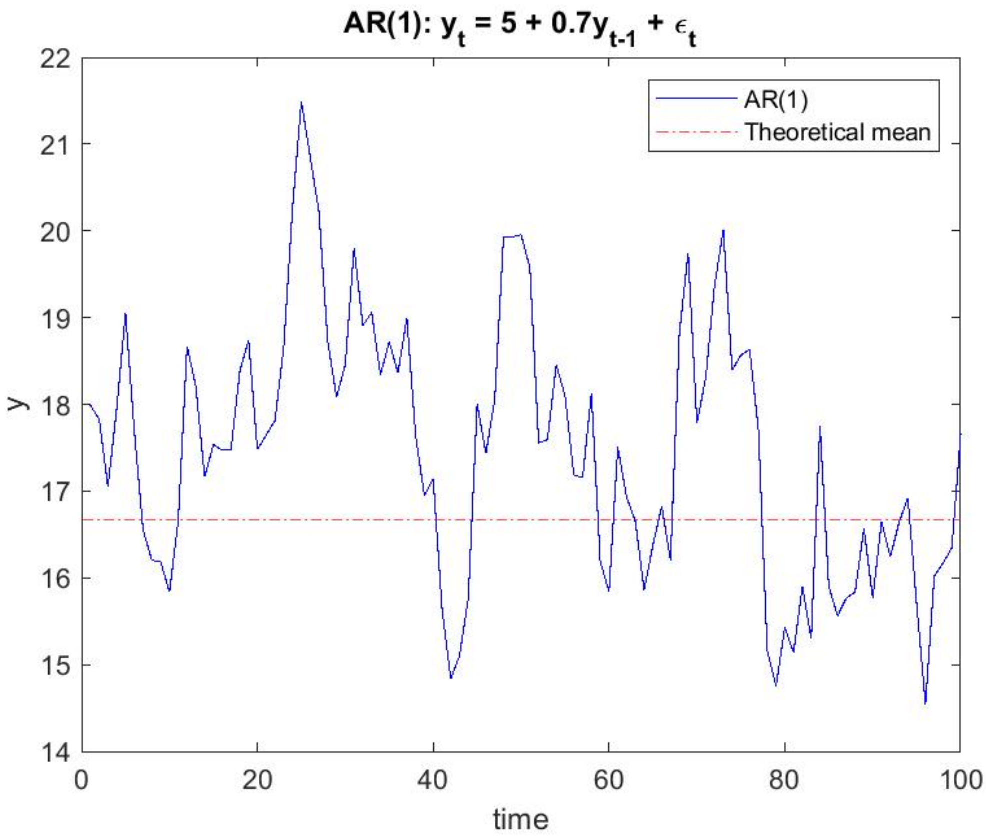

Figure 1: *AR(1) process*

Since $\epsilon_t \overset{i.i.d.}{\sim} N(0,1)$ and $|\phi| < 1$, the formula in equation 2 can be used to calculate the theoretical mean which is drawn in the graph.

$$E[Y_t] = \frac{c}{1-\phi} \tag{2}$$

The formula results in a theoretical mean of 16.6667. Since the AR(1) process in the figure seems to have a negative trend (which would contradict the assumption of stationarity) we controlled the process for a higher number of observations (T=500). This shows that in the long-run, the AR(1) oscillates around the theoretical mean as expected. The result is shown in appendix A.

2.2 Conditional Log Likelihood Function

In section 2.1, we already verified that our time series fulfills the necessary requirements for the maximum likelihood. We will condition on the first observation y_1 and take the log, which greatly simplifies the computation:

$$ln\ \mathcal{L} = \sum_{t=2}^{T} ln\Big(\frac{1}{\sqrt{2\pi\sigma^2}}\ exp\Big[\frac{(y_t - c - \phi y_{t-1})^2}{-2\sigma^2}\Big]\Big) = \sum_{t=2}^{T}\Big[ln\Big(\frac{1}{\sqrt{2\pi\sigma^2}}\Big) - \frac{\epsilon_t^2}{2\sigma^2}\Big], \qquad (3)$$

with $\epsilon_t = y_t - c - \phi y_{t-1}$, σ^2 the variance of Gaussian White Noise innovation terms and the AR(1) process $y = (y_1, y_2, ..., y_{100})'$ described in section 2.1. The input variables for the function are $\theta = (c, \phi, \sigma^2)$ and the time series y generated with equation (1). Summing up the likelihood contributions yields the Conditional Log Likelihood function. Two different sets of input variables are tested with the following results:

a.) θ = (5, 0.7, 1)' with: $ln\ \mathcal{L} = -139.0975$

b.) θ = (-5, 0.1, 0.25)' with: $ln\ \mathcal{L} = -85467$

As expected, the chosen parameters in a.) describe the time series more accurately, which can be identified by the higher value of $ln\ \mathcal{L}$. This result is perfectly reasonable as the parameters in a.) are identical to the true parameters that generated the original series.

3 Estimation of the AR(1) process parameters

To estimate the true parameters of the AR(1) process, different methods are applied on two different starting point settings. The settings use the parameter of a.) and b.) described in section 2.2. The three methods applied for the estimations are:

- *fminunc*: Finds minimum of unconstrained multivariable function[1]
- *fminsearch*: Finds minimum of unconstrained multivariable function using derivative-free method[1]
- *patternsearch*: Finds minimum of function using pattern search[1]

All three methods are tested for both settings and deliver the following results:

		fminunc	**fminsearch**	**patternsearch**
a.)	c	4.0220	4.0220	4.7109
	ϕ	0.7697	0.7697	0.7303
	σ^2	0.9036	0.9036	0.9072
b.)	c	4.0215	4.0220	15.0000
	ϕ	0.7697	0.7697	0.1469
	σ^2	0.9042	0.9036	1.7813

Table 1: *Estimation results*

Table 1 shows that the estimation results for *fminunc* and *fminsearch* deliver quite similar results, independent of the starting point setting. For *patternsearch*, the precision of the result seems to depend on the starting value of the estimation function. In case the starting point is quite close to the true values, *patternsearch* is able to estimate the parameters very precisely. As soon as the starting values move further away from the true value, like in parameter setting b.), the estimation results become worse.

[1]The definitions are from the MATLAB function documentation MATLAB (b), MATLAB (a) and MATLAB (c)

These results can also be visualised using a Log Likelihood profile. Figures 2 and 3 describe the Log Likelihood profiles for each parameter setting and each estimation method over a varying range of ϕ holding the other parameter fix[2]. The blue line shows the Log Likelihood profile, the red line shows the estimated $\hat{\phi}$ and the black one shows the true value of the AR(1) process. With a perfect estimation, the black line (true value) would line up with the red line in the maximum of the Log Likelihood profile.

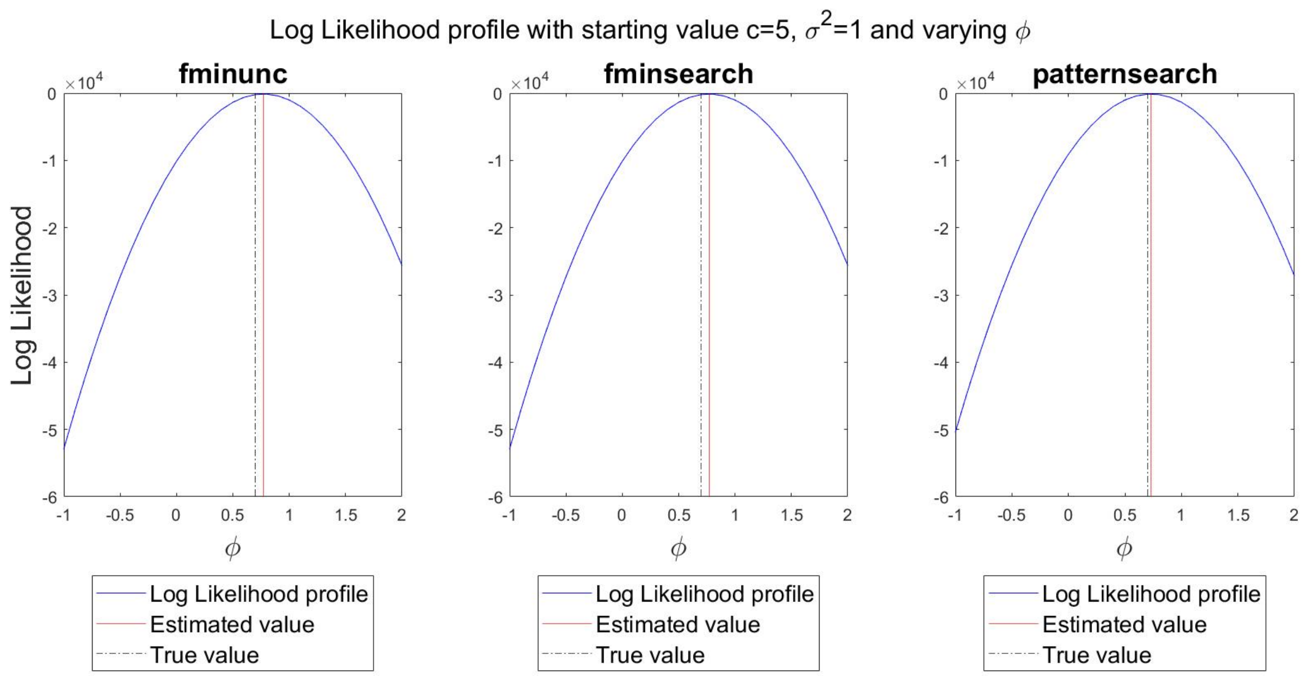

Figure 2: *Log Likelihood profile using parameters a.) as starting values*

As visible in figure 2 all three methods estimate the parameters quite precisely. There are only marginal differences between the three methods and they all give a reasonable estimate. However, it is not a realistic scenario that the starting values are equal to the true value. It is more realistic to have a different starting value, as it is the case in b.).

[2]The chosen range of ϕ in the graphs is $[-1; 2]$. We expect ϕ to be in range $[-1; 1]$ but for a better overview of the Log Likelihood profile we extended the range to 2.

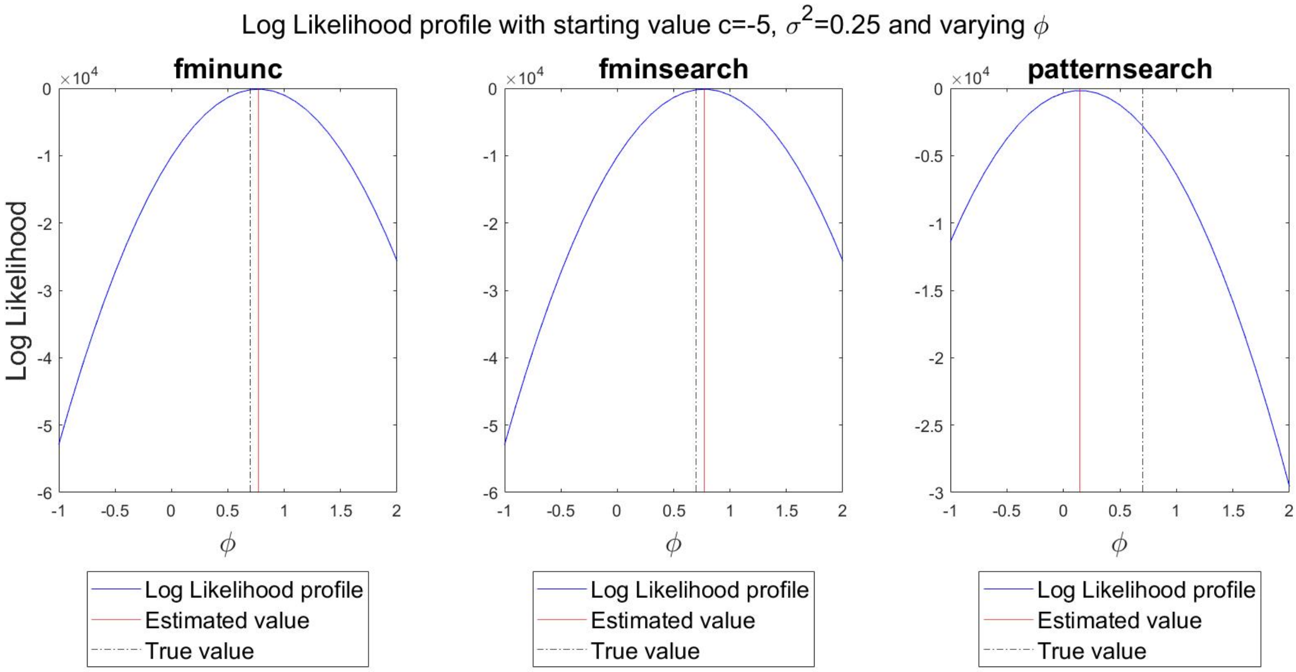

Figure 3: *Log Likelihood profile using parameters b.) as starting values*

As already described by the results in table 1, *fminunc* and *fminsearch* deliver quite precise results even if the starting point differs substantially from the true values. For *patternsearch* it shows that the estimation function becomes less precise the further we move away from the true values. This is shown by the large distance between the line for the true value and the estimated value for *patternsearch* in figure 3.

4 Statistical inference

In this part, statistical inference is conducted by using the provided CML toolbox. As displayed in table 2, the parameters calculated in section 3 are identical to the results of the CML toolbox.

		fminsearch	**CML toolbox**
a.)	c	4.0220	4.0220
	ϕ	0.7697	0.7697
	σ^2	0.9036	0.9036
b.)	c	4.0220	4.0220
	ϕ	0.7697	0.7697
	σ^2	0.9036	0.9036

Table 2: *Comparison of the fminsearch function with the CML toolbox*

The toolbox allows us to use the three different optimization algorithms and also gives us a covariance matrix either based on the inverse of the Hessian matrix, the inverse of the cross-product of the first derivatives or the Quasi Maximum Likelihood covariance matrix.

4.1 Statistical Inference of a single realization

To analyze the performance of the Maximum Log Likelihood estimation, we test H_0 that $\phi_0 = 0.7$. This is tested with the t-statistic and a significance level of $\alpha = 0.05$. The t-statistic looks as follows:

$$t = \frac{\hat{\phi} - \phi_0}{\hat{\sigma}} \tag{4}$$

We get the estimated standard error s.e. ($\hat{\sigma}$) from the covariance matrix that is based on the inverse of the Hessian matrix. The value of the t-statistic is 1.0876, which is smaller than the critical value of $z_{[1-\frac{\alpha}{2}]} = 1.96$ (we can use the z-value since T is large enough). We therefore cannot reject the null-hypothesis that ϕ_0 is equal to 0.7, meaning that the difference from our estimation to our true value of ϕ is not statistically significant. This can also be seen by looking at the confidence interval that is calculated using equation (5):

$$CI = \left[\hat{\phi} - z_{[1-\frac{\alpha}{2}]}\hat{\sigma}\ ;\ \hat{\phi} + z_{[1-\frac{\alpha}{2}]}\hat{\sigma}\right] \tag{5}$$

The result using this formula is the following confidence interval containing all values that are not rejected at the 5% significance level, which includes the value of 0.7:

$$CI = [0.64409\ ;\ 0.89532]$$

Using the inverse of the cross-product of the first derivatives and the Quasi ML, the standard errors are calculated again. In theory, the results should be the same, however, they differ. We increase T from 100 to 50000, which is expected to lead to a higher accuracy. Conducting the hypothesis test again, we now get a very small t-statistic of 0.49458, showing that the estimation is very close to the true value. Also, when looking at the confidence interval now, the difference between the upper and lower bound is much smaller, which also shows that we have a higher accuracy.

$$CI = [0.69215\ ;\ 0.70469]$$

In addition, the standard errors calculated with the three different methods are now very close to each other. An overview over the standard errors for T=100 and T=50000 is provided in appendix B.

4.2 Statistical Inference of an Ensemble

We generate the AR(1) process from equation 1 again but with T = 300. We simulate this process for 100 ensembles, meaning that we restart the world 100 times. In reality, this is not possible as we observe a time series only once but having an experimental environment allowing us to restart the world would be an ideal situation. With this information, we can calculate the confidence interval for multiple estimators of the same process and check if the true value of $\phi = 0.7$ is in the rejection area or not. In our case, 94 out of 100 realizations have 0.7 in the confidence interval. As expected, in almost all cases the true value lies within the confidence interval.

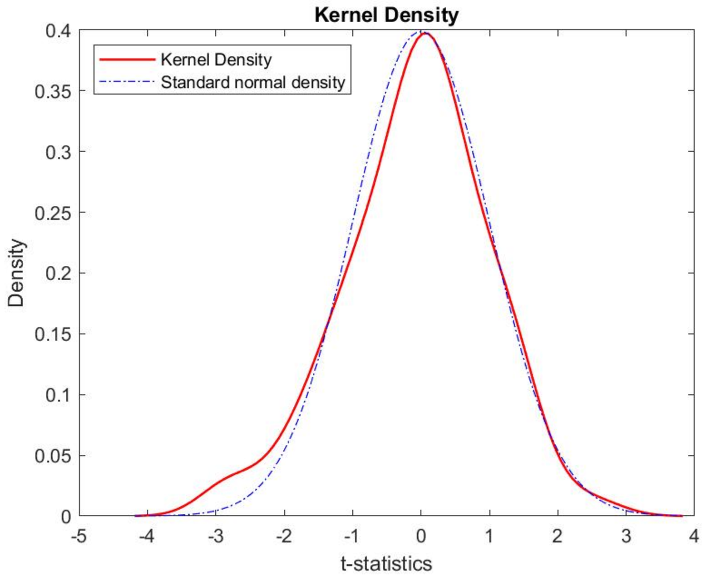

Figure 4: *Kernel density of the t-statistic*

The kernel density of the t-statistic from the 100 ensembles is plotted together with a standard normal distribution in figure 4. As T is large enough, the distribution of the t-statistic converges to the standard normal distribution, which is clearly visible in the graph.

5 Conclusion

In this short paper, we were able to show that the conditional Log Likelihood is an efficient and consistent estimator if the requirements of stationarity, ergodicity and the correct specification of the joint distribution are fulfilled. *fminunc* and *fminsearch* lead to almost identical estimations of the parameter while *patternsearch* relies on a starting point that is close to the true value to give a good estimation. An increasing T leads to a higher accuracy of the estimations and a narrower confidence interval, as expected by a consistent estimator. Also, when simulation the ensemble with 100 repetitions, the distribution of the t-statistic converged to a standard normal distribution, hence on average the estimated ϕ converges to the true value.

References

Hamilton, J. D. (1994). *Time Series Analysis*, volume 2. Princeton University Press Princeton, NJ.

MATLAB. fminsearch Function Documentation. `https://de.mathworks.com/help/matlab/ref/fminsearch.html`. Last accessed: 2019-11-28.

MATLAB. fminunc Function Documentation. `https://de.mathworks.com/help/optim/ug/fminunc.htm`. Last accessed: 2019-11-28.

MATLAB. patternsearch Function Documentation. `https://de.mathworks.com/help/gads/patternsearch.html`. Last accessed: 2019-11-28.

A AR(1) process with T=500

To check the long-run development of the AR(1) process in section 2.1 we increased the number of observations to T=500.

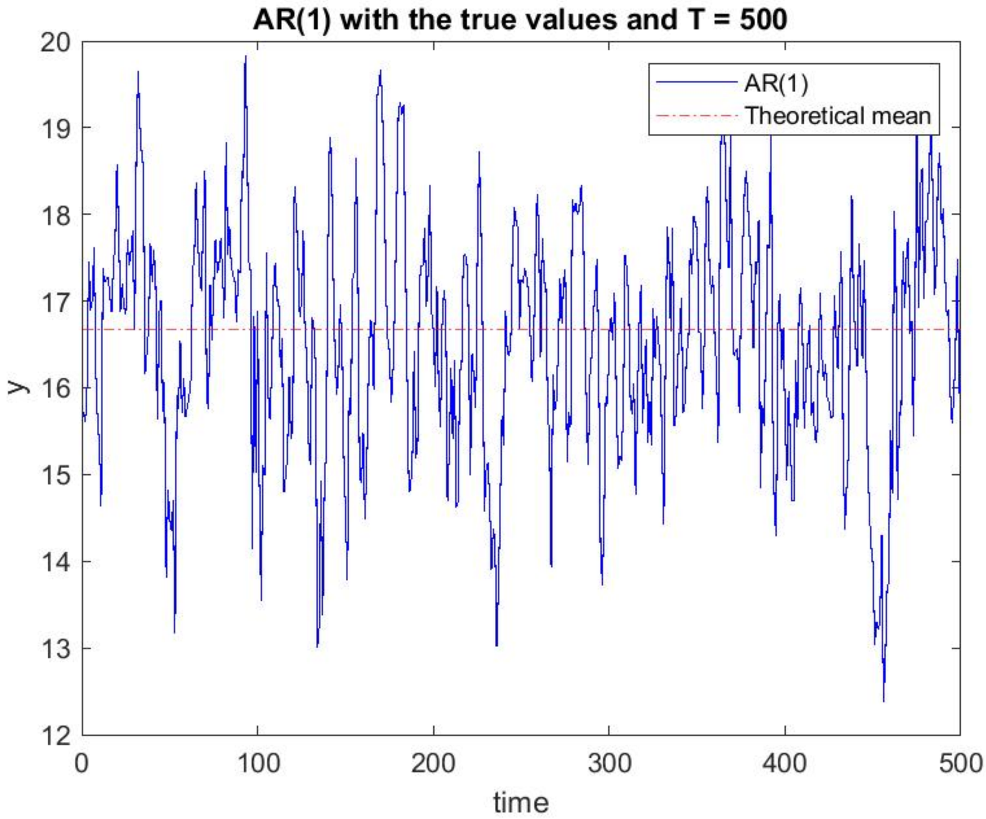

Figure 5: *AR(1) process with T=500*

Figure 5 shows that the AR(1) oscillates around the theoretical mean and also seem not to vary in distribution over time. This supports the assumptions of stationarity and ergodicity.

B Standard Errors of all the estimates

Methods used to determine the standard errors of the parameter estimation:

- Method 1: Inverse of Hessian
- Method 2: Inverse of cross-product of derivatives
- Method 3: Quasi-ML covariance matrix

	Method 1	**Method 2**	**Method 3**
		T = 100	
se(c)	1.1243	1.2045	1.0629
se(ϕ)	0.0641	0.0684	0.0608
se(σ^2)	0.1284	0.1366	0.1231
		T = 5000	
se(c)	0.0535	0.0530	0.0539
se(ϕ)	0.0032	0.0032	0.0032
se(σ^2)	0.0063	0.0063	0.0063

Table 3: *Overview over the standard errors of the estimates using different methods*

As expected, an increase in T leads to smaller standard errors and eliminates the differences between the methods applied to get the standard errors of the estimates.

C List of Variables

Table 4: *List of Variables - Task 1 to 3.2*

Name	Description	Created in task
ar1	AR(1) process	1
theo__mean	Theoretical mean of the AR(1) process	1
X	X-axis (1 to 100) for the plot	1
par__a	Parameter vector for setting a)	2.2
par__b	Parameter vector for setting b)	2.2
log__likelihood__a	Log Likelihood using a)	2.2
log__likelihood__b	Log Likelihood using b)	2.2
func__a	Evaluation function for a)	3.1
func__b	Evaluation function for b)	3.1
max__fminunc__a	Optimization fminunc for a)	3.1
max__fminsearch__a	Optimization fminsearch for a)	3.1
max__patternsearch__a	Optimization patternsearch for a)	3.1
max__fminunc__b	Optimization fminunc for b)	3.1
max__fminsearch__b	Optimization fminsearch for b)	3.1
max__patternsearch__b	Optimization patternsearch for b)	3.1
temp__logli__vec__a	Log Likelihood vector for a) (temporary)	3.2
temp__logli__vec__b	Log Likelihood vector for b) (temporary)	3.2
range__phi	Range of ϕ which should be tested	3.2
c__est__a	Vector with estimated c's using parameters of a)	3.2
var__est__a	Vector with estimated variances using parameters of a)	3.2
c__hat__a	Scalar holding the c used in the respective loop for the a) setting	3.2
var__hat__a	Scalar holding the variance used in the respective loop for the a) setting	3.2
temp__par__a	Vector holding parameter for a) (temporary)	3.2
temp__logli__scalar__a	Scalar holding the Log Likelihood for a) (temporary)	3.2
matrix__a	Matrix holding the Log Likelihood values for all ϕ for a)	3.2
c__est__b	Vector with estimated c's using parameters of b)	3.2
var__est__b	Vector with estimated variances using parameters of b)	3.2
c__hat__b	Scalar holding the c used in the respective loop for the b) setting	3.2
var__hat__b	Scalar holding the variance used in the respective loop for the b) setting	3.2
temp__par__b	Vector holding parameter for b) (temporary)	3.2
temp__logli__scalar__b	Scalar holding the Log Likelihood for b) (temporary)	3.2
matrix__b	Matrix holding the Log Likelihood values for all ϕ for b)	3.2

Table 5: *List of Variables - Task 4.1 to 4.6*

Name	Description	Created in task
cml_est_fmin_a	Toolbox estimation fminsearch for a)	4.1
cml_cov_hes_a	Toolbox covariance hessian for a)	4.1
cml_est_fmin_b	Toolbox estimation fminsearch for b)	4.1
cml_cov_hes_b	Toolbox covariance hessian for b)	4.1
se_hessian_vec	Vector holding the standard errors of the estimations using the true values	4.2
test_stat	Scalar holding the test statistic for the ϕ estimator	4.2
conf_interval_low	Scalar holding the lower bound of the confidence interval	4.3
conf_interval_upper	Scalar holding the upper bound of the confidence interval	4.3
cml_est_fmin_OPG	Toolbox estimation fminsearch using inverse of the cross-products of the inverse	4.4
cml_est_fmin_QML	Toolbox estimation fminsearch using quasi-ML covariance matrix	4.4
se_OPG_vec	Scalar holding the standard error of the ϕ estimation using the inverse of the cross-products of the inverse	4.4
se_QML_vec	Scalar holding the standard error of the ϕ estimation using quasi-ML covariance matrix	4.4
ar1_alt	Alternative AR(1) process with T=5000	4.5
cml_est_fmin_alt	Toolbox estimation fminsearch using the alternative AR(1) process	4.5
cml_cov_hes_alt	Toolbox covariance hessian for the alternative AR(1) process	4.5
se_hessian_vec_alt	Vector holding the standard errors of the estimations using the alternative AR(1) process	4.5
test_stat_alt	Scalar holding the test statistic for the ϕ estimator for the alternative AR(1) process	4.5
conf_interval_low_alt	Scalar holding the lower bound of the confidence interval for the alternative AR(1) process	4.5
conf_interval_upper_alt	Scalar holding the upper bound of the confidence interval for the alternative AR(1) process	4.5
cml_est_fmin_OPG_alt	Toolbox estimation fminsearch using inverse of the cross-products of the inverse for the alternative AR(1) process	4.5
cml_est_fmin_QML_alt	Toolbox estimation fminsearch using quasi-ML covariance matrix for the alternative AR(1) process	4.5
se_OPG_vec_alt	Scalar holding the standard error of the ϕ estimation using the inverse of the cross-products of the inverse for the alternative AR(1) process	4.5
se_QML_vec_alt	Scalar holding the standard error of the ϕ estimation using quasi-ML covariance matrix for the alternative AR(1) process	4.5
matrix_ensemble	Matrix holding the t-statistic and the CI of each of the 100 ensembles of the original AR(1) process	4.6
ar1_ens	AR(1) process of the respective iteration	4.6
cml_est_fmin_ens	Toolbox estimation fminsearch for the respective iteration	4.6
cml_cov_hessian_ens	Toolbox covariance hessian for the respective iteration	4.6
test_stat_ens	Scalar holding the test statistic for the ϕ estimator for the respective iteration	4.6
conf_interval_low_ens	Scalar holding the lower bound of the confidence interval for the respective iteration	4.6
conf_interval_upper_ens	Scalar holding the upper bound of the confidence interval for the respective iteration	4.6