Lukas Humpe

The Properties of AR(1) Models: A Simulation Study

Term Paper
Winter Term 2019

Submitted to
Institute of Economics and Statistics at the University of Cologne
Cologne (2020)

Table of Contents

List of Figures

List of Tables

1. Autoregressive Time Series

Time series analysis is useful in manifold applications. It allows for the prediction of future values, inferences made on the structure of processes and can therefore be used as a decision supporting tool in businesses, research and economics (Huitema & McKean, 1991; Lawer, 2016). The aim of most time series models is to remove seasonal and trend related effects, then model time related dependency structures that are present in order to fit the often imperfectly predictable residuals to be able to reverse the transformations to create predictions. Autoregressive Integrated Moving Average (ARIMA) models of Box and Jenkins (1970) are amongst the most popular models to apply this form of time series analysis (McCleary et al., 2017). One of the building blocks of these models are autoregressive (AR) models which model autocorrelation between past and future observations. In order to be able to identify the correct model parameters and to make inferences about time series, that are governed by an AR process, one needs to understand the underlying assumptions and dynamics of time series in general as well as the specific properties of AR processes and models.

A discrete univariate time series is a collection of observations depicting measurements of one variable. In contrast to the continuous case, where there is a measurement for every moment in time, measurements in discrete time series are either equally or unequally spaced (Brockwell & Davis, 2016, p. 1). For the scope of this paper, the term time series is used to describe a discrete univariate time series with equally spaced time intervals. A time series can be viewed as an ordered sequence of random variables $\{Y_1, ..., Y_T\}$ and their realization $\{y_1, ..., y_T\}$. One assumption is that time series could in principle be observed from infinite past to the infinite future. In reality however only samples are available. In addition, only one specific realization of all that are possible can be examined (e.g. there is only one available realization of the German stock index). A sample with size T of a specific realization $\{y_1, ..., y_T\}$ is called a trajectory of Y_t. In theory, time series therefore resemble stochastic processes, when the possibility of different trajectories is taken into account and are therefore only imperfectly predictable. When modeling such time series one has to find the specific process that governs the trajectory of the time series. One property present in many time series is autocorrelation meaning that values of Y_t at time t are correlated with Y_t at some other time $t - j$, where j depicts the distance between two points in time. The process underlying such time series is called AR process. An AR process in which Y_t is correlated with the last p past values is called an AR process of order p ($AR(p)$ in shorthand notation). One of the most popular processes found in practice are first-order AR processes $AR(1)$ satisfying the following difference equation.

$$Y_t = c + \phi_1 Y_{t-1} + \epsilon_t \qquad \text{with } \epsilon_t \sim N(0, \sigma^2) \tag{1}$$

where ϕ_1 depicts the amount of autocorrelation between Y_t and Y_{t-1} and ϵ_t is a sequence of

independent and identically distributed random variables $\{\epsilon_1, \ldots, \epsilon_t\}$ with $\epsilon_t \sim N(0, \sigma^2)$ called Gaussian white noise (GWN). Note that the intercept c will not be considered in the rest of this paper as it can be easily eliminated by subtracting c from both sides. Due to the stochastic nature of AR processes to be able to make inferences about the process one needs to calculate the stable moments of the process. A necessary condition to obtain these is weak or covariance stationarity. A time series $\{Y_t\}$ with t $\in \mathbb{Z}$ and $E(Y_t^2) < \infty$ is weakly stationary if the following conditions hold:

$$E(Y_t) = \mu \qquad \text{for all t} \tag{2}$$

$$E[(Y_t - \mu_t)(Y_{t-j} - \mu_{t-j})] = \gamma_j \qquad \text{for all t and each j} \tag{3}$$

In general it can be said that all these conditions ensure that the moments are stable over time i.e. that all elements of the trajectory are governed by the same distribution so that inferences about the properties can be made beyond the single sample. Equation 2 hereby ensures that the expected value is stable over time. Equation 3 states that the autocovariances do not depend on t but only on the distance j of the two points in time. This is necessary to ensure that ϕ for a specific j is constant over time but is allowed to vary for different distances. The above would not be the case if there were any kind of trend, seasonality or structural breaks present in the process. As a trend for example would lead to a different μ depending on t and seasonality would result in varying autocorrelation. In practice these effects therefore have to be filtered before modelling the time series. The specifics of these techniques however are beyond the scope of this paper and only stationary processes will be considered. To show that an $AR(1)$ process with $|\phi_1| < 1$ is stationary one has to recursively solve Equation 1 for Y_{t-j} which yields the following result.

$$Y_t = \epsilon_t + \phi\epsilon_{t-1} + \phi^2\epsilon_{t-2} + \cdots + \phi^j\epsilon_{t-j} + \phi^{j+1}Y_{t-j-1} \tag{4}$$

$|\phi_1| < 1$ is necessary because $\phi^{j+1}Y_{t-j-1}$ will vanish as $j \to \infty$, for $|\phi_1| \geq 1$ the following result does therefore not hold.

$$Y_t = \sum_{j=0}^{\infty} \phi^j \epsilon_{t-j} \tag{5}$$

By taking the expectation from both sides of Equation 5 together with the fact that $E(\epsilon_t) = 0$ per definition one can see that $E(Y_t) = 0$ for all t.

$$E(Y_t) = \mu = \sum_{j=0}^{\infty} \phi^j E(\epsilon_{t-j}) = 0 \tag{6}$$

Showing that $E(Y_t)$ is independent of t. The autocovariance can then be calculated as

$$\gamma_j = Cov(Y_t, Y_{t-j}) = E[(Y_t)(Y_{t-j})] = \frac{\sigma^2 \phi^j}{1-\phi^2} \tag{7}$$

It is easy to see that γ_j does not depend on t therefore the conditions of weak stationarity are met. To get the autocorrelation of the process the autocovariance is standardized by the variance γ_0. The autocorrelation function (ACF) is then given as

$$\rho_j = \frac{\gamma_j}{\gamma_0} = \frac{\frac{\sigma^2 \phi^j}{1-\phi^2}}{\frac{\sigma^2 \phi^0}{1-\phi^2}} = \phi^j \tag{8}$$

As a result of Equation 8 it is intuitive that the autocorrelation of an $AR(1)$ process with $|\phi_1| < 1$ decays very quickly when increasing j but is not zero for $j > 1$. This due to the fact that autocorrelation at $j > 1$ is not a result of Y_t being correlated with Y_{t-j} but rather Y_t being correlated with Y_{t-1} which is correlated with Y_{t-j}. The serial correlation of two points in time is therefore carried forward throughout the time series getting insignificant after some time. To find the true autocorrelation in an $AR(p)$ process for all lags, the autocorrelation of Y_{t-j} has to be calculated after having accounted for all previous autocorrelations, which results in the partial autocorrelation function (PACF). The PACF is especially important when choosing the order of an AR process, since it correctly reflects the structure of serial correlation in the processes. In the case of an $AR(1)$ process however there are no previous autocorrelations that have to be accounted for and the first autocorrelation and partial autocorrelation are equal. As this paper deals only with $AR(1)$ processes the PACF will not be derived.

Since nothing can be known about the process in reality, inferences about the process can only be made using estimates (specific calculations will be shown in section 3). To successfully build a model that utilizes these estimates one has to determine the lag order of the process, which can be done by examining PACF plots. Under certain circumstances however this can be difficult even if the process is stationary. Figure 1 shows the estimated PACF of an $AR(1)$ process with $\phi = 0.8$ and $\epsilon_t \sim N(0, 1)$ and a sample size of $T = 25$. It can be seen that at lag one the autocorrelation was not estimated precisely, since the spike at lag one shows a value just above 0.6. Also the autocorrelation at lag nine was estimated to be significant. If one were to build a model out of this it would result in misspecification since the wrong order p would be selected as well as wrong values for ϕ_j. This example therefore shows that model estimates regarding the strength of autocorrelation as well as the lag order of an AR process can vary under certain conditions. To examine under which conditions precise $AR(1)$ models can be estimated, in the remainder of this paper a simulation study will be conducted. Section 2 will explain the method after which section 3 introduces the techniques used to estimate the desired parameters. In section 4 the results will be presented which will be discussed and concluded in the last two sections.

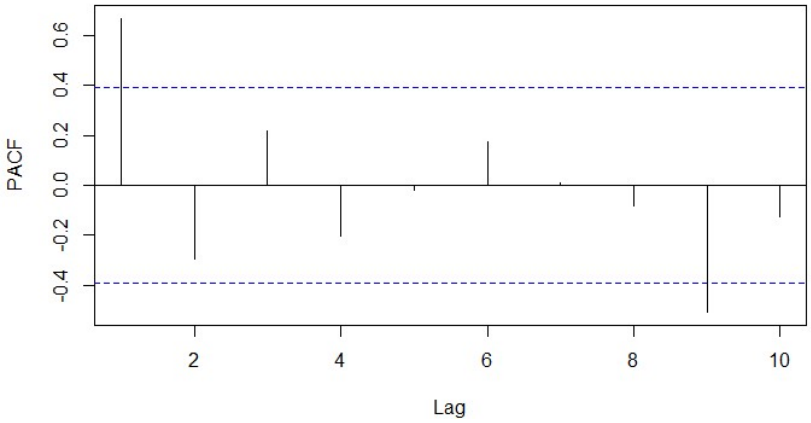

Figure 1: Deceptive PACF Plot of an AR(1) Process

2. Method

To study the properties of $AR(1)$ models a Monte-Carlo study was conducted. In this the length T of the time series and the theoretical autocorrelation ϕ will be varied. For ϕ, 19 different autocorrelations will be studied from a range of -0.9 to 0.9 in steps of 0.1. Values of $|\phi| \geq 1$ are excluded for reasons indicated in section 1 and since values of $1 \geq \phi \leq -1$ cannot be estimated because the resulting autocorrelation matrix must be positive definite (Scheffé, 1959, p. 334) to be able to derive the inverse. Positive and negative values for ϕ will be included as previous studies have found a difference in bias between positive and negative values of ϕ (Solanas et al., 2010). For T there will be five different values, namely 25, 50, 100, 500 and 1000. These values were chosen so that the whole bandwidth i.e. very small values as well as particularly large data sets are included. Even though smaller data sets are more common in practice it can be expected that with current development and technology trends such as big data larger data sets are getting more prominent in the future. For each parameter combination 10000 replications have been conducted resulting in 19 x 5 x 10000 = 950000 simulations. Each AR process was simulated in R (Version 3.6.3) and the first 100 observations were dropped to eliminate the influence of the initialization step. μ was set to 0 and σ_e^2 was set to 1 for all replications as this can be done without losing generality (Krone et al., 2017). Code for all custom functions used and the whole simulation process as well as code for graphics are provided in the appendix A.1, to ensure that the results are reproducible.

3. Estimation

The autocorrelation coefficient $\hat{\phi}$ is estimated using the most frequently utilised (Huitema & McKean, 1991; Solanas et al., 2010) r_1 estimator:

$$\hat{\phi} = r_1 = \frac{\sum_{t=1}^{n-1}(Y_t - \bar{\mu})(Y_{t+1} - \bar{\mu})}{\sum_{t=1}^{n}(Y_t - \bar{\mu})^2} \tag{9}$$

where $\bar{\mu}$ is the sample mean of all observations. It is important to note that this estimator resembles the empirical ACF of equation Equation 8 for $j = 1$ which only gives precise estimates for the first autocorrelation coefficient of an $AR(p)$ process. For the estimation of AR processes with orders $p > 1$ other estimators would have to be utilized (e.g. Yule-Walker equations or Ordinary Least Squares(OLS)) since only $AR(1)$ processes are investigated this estimator however is proficient. In addition the r_1 estimator does not violate as many assumptions compared to e.g. the OLS estimator where the basic assumption that the error terms are independent of all entries of the regressors is violated (due to the fact that y_t is per definition dependent on y_{t-j} which is dependent on ϵ_t). To examine how far away $\hat{\phi}$ is from ϕ in the mean, the bias will be recorded. The bias will be computed as done by other studies (Arnau & Bono, 2001; Krone et al., 2017)

$$Bias = \left(\frac{1}{N}\sum_{i=1}^{N}\hat{\phi}_i\right) - \phi \tag{10}$$

where N denotes the number of replications. To observe the overall variability of $\hat{\phi}$ the empirical standard error will be examined. It is computed as follows:

$$SE(\hat{\phi}) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{\phi}_i - \bar{\hat{\phi}}\right)^2} \tag{11}$$

This will be used as a proxy for the confidence of estimation since high variability would suggest that estimates are less reliable.

4. Results

In the following, results will be presented only graphically. Precise numerical results will be used in the discussion and are provided in Table 1 in section A.2.

4.1. Estimation and Bias

Figure 2 illustrates the mean of $\hat{\phi}$ and the associated mean bias related to samples sizes and different values for ϕ. At first glance it becomes clear that estimates of ϕ get less biased the higher the number of observations. Another finding that catches the eye is that estimates are most precise when ϕ takes a value of -0.3, as all curves intersect in that point approximately. Additionally it can be observed that the bias seems to be positive when $\phi \leq -0.3$ and is negative when $\phi > -0.3$. This however is less prominent when working with sample sizes $T \geq 500$. In addition improvements in accuracy seem to be larger when increasing the number of observations from 25 to 50 than increasing T from 500 to 1000, which can be seen comparing the reduction in bias for different values of T.

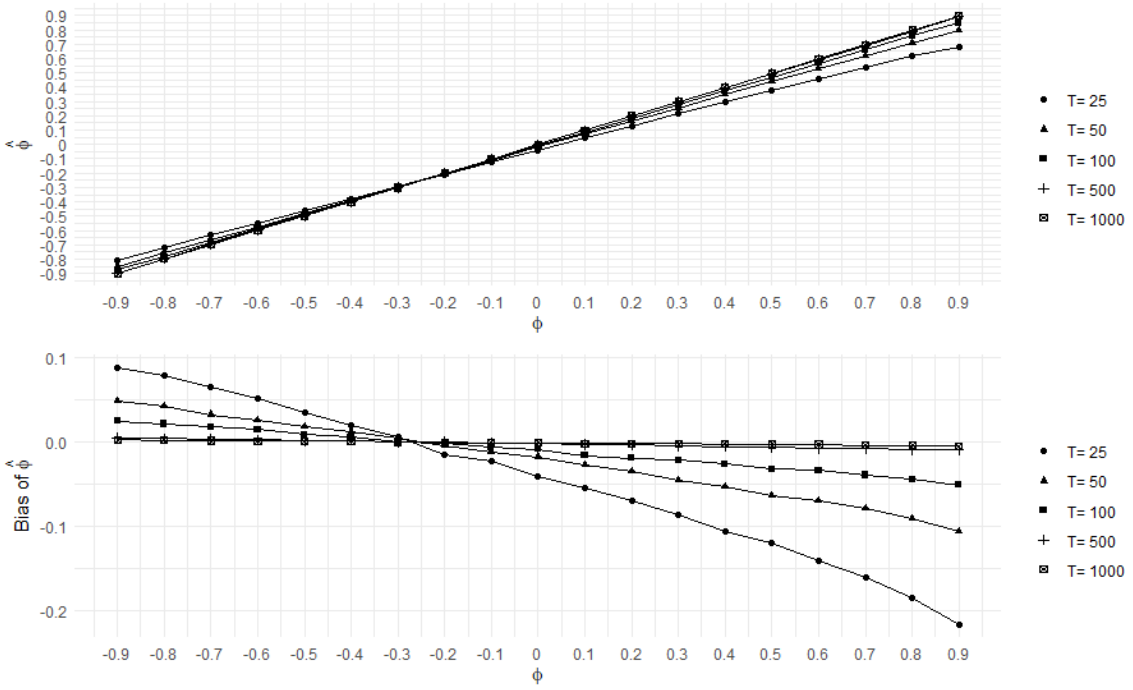

Figure 2: Estimates and Bias

4.2. Variability

Figure 3 shows the empirical standard error for different values of ϕ and different sample sizes.

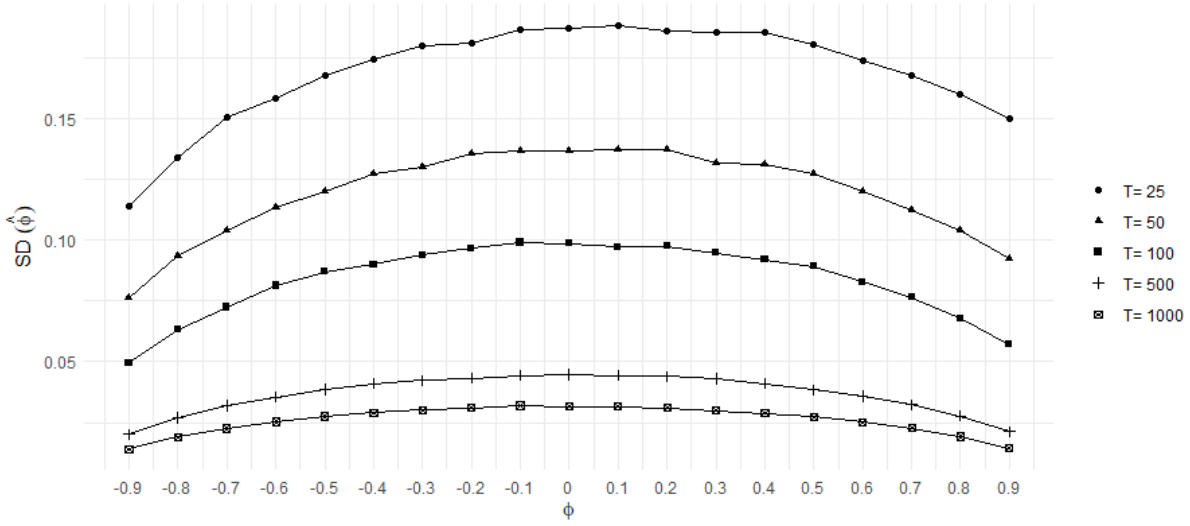

Figure 3: Empirical Standard Error of $\hat{\phi}$

The main finding is that standard error is the highest for samples of size $T = 25$ for a respective value of ϕ and least for $T = 1000$. Another prominent result is that the standard error seems to be largest for $\phi = 0$ and lowest for $\phi = -0.9$, as it increases from $\phi = -0.9$ to $\phi = 0$ and then decreases again however not as much as they were increasing before resulting in a smaller standard error for positive values of ϕ compared to their

negative counterparts. In addition the standard error improves more when increasing small sample sizes compared to increasing the number of observations of large sample sizes which can be seen from larger distances between curves for $T = 25$ and $T = 50$ compared to $T = 500$ and $T = 1000$. Furthermore curves for large sample sizes seem to be more flat compared to small sample sizes.

5. Discussion

In this simulation study the consistency of the r_1 estimator was examined in respect to variability and mean bias when varying sample sizes and values of ϕ. The main intention was to examine and unfold the properties of $AR(1)$ processes to be able to study and compare them to estimated $AR(1)$ models to improve insights for the order selection process when modelling $AR(1)$ time series and thus to avoid misspecification. The results of this study confirmed findings of previous studies to a large extend.

Since bias and variability were highest for small sample sizes it can be concluded that estimation and model selection using small samples can result in misspecification of the model. This can be in terms of order p as well as ϕ as already illustrated in Figure 1. To what extend a misspecification in respect to the lag order is possible is however beyond the scope of this paper. At samples with $T \geq 500$ however, curves flattened considerably with $0.000 \leq Bias \leq 0.010$. For $\phi = -0.3$ bias was lowest along all sample sizes with $Bias = 0.000$ for $T \geq 100$ and $Bias \leq 0.006$ for $T \leq 50$. For $0.3 < \phi < -0.3$ observable bias was prominent for all sizes of T. Increasing more rapid for $0.3 < \phi$ than for $\phi < -0.3$. More specifically it could be observed that r_1 overestimates for $\phi < -0.3$ and underestimates for $\phi > -0.3$ which was also confirmed by Arnau and Bono (2001, p. 368). This can be explained by the fact that relevant statistics are calculated with the assumption of independence not taking into account the serial correlation (Scheffé, 1959, p. 331 ff.). In practice however this does not make a lot of difference, since the value of the true ϕ cannot be known before hand. Thus one cannot argue that estimates are completely reliable just because they are close to 0.3. It can however be argued that these values are more trustworthy compared to very high values of ϕ to a certain extend. Nonetheless, this cannot be generalized for the variability of $\hat{\phi}$ as the empirical standard error was lowest for $\phi = -0.9$ increasing until $\phi = 0$ and then decreasing again with $SD(\hat{\phi})$ being smaller for $\phi = -0.9$ than for $\phi = +0.9$. This implies that variability is higher for positive values of ϕ than for negative. This held true for all sizes of T except for $T = 1000$ where $SD(\hat{\phi})$ was approximately symmetric. In general it was observed that $SD(\hat{\phi})$ was lower for larger sample sizes. It is important to note that this is not a result of dividing by T as the empirical standard error was derived in respect to the number of replications as defined in Equation 3. It can therefore be argued that apart from being the most precise in terms of bias larger samples result in lower variability of the estimates. As with the bias of $\hat{\phi}$ improvements in $SD(\hat{\phi})$ were getting lower the further T was increased. Therefore, it

got clear that the estimation of $\hat{\phi}$ is not only depended on the model parameters selected by the practitioner but also on the non modeled parameters such as size of available data.

It is important to note that statements made in this paper cannot be generalized to $AR(p)$ models, or even other models such as Moving average models, since only $AR(1)$ models were employed to simulate the time series. In addition the simplest form of estimator was chosen in this study. Even though the r_1 estimator is among the most heavily utilised estimators, there are already various different estimators that aim to reduce issues associated to the r_1 estimator. The investigation of these was however beyond the scope of this study. In addition it was found by other studies that also the standard error is biased (Arnau & Bono, 2001; Solanas et al., 2010) which indicates that it even more caution is required since most of the derived statistics are used within many forecasting and intervention scenarios as well as for hypothesis testing (Huitema & McKean, 1991). Since properties such as rejection rate or bias of the empirical standard error were not considered, it is important to note that the findings of this study cannot be readily be generalized.

6. Conclusion

In general it can be said that high sample sizes as well values for ϕ close to -0.3 yield the more trustworthy results. The results further suggest that a sample with size $T = 500$ is already quite proficient for the reliable estimation of ϕ in centered, stationary $AR(1)$ processes, as more data does not seem to improve variability and estimation precision by much. Another main result is that practitioners have to keep in mind that whether bias is positive or negative is dependent on amount of autocorrelation present in the process. This insight however on its own does not really allow to make inferences. For small samples that show high estimated autocorrelation it is therefore advised to test the residuals of the model to check whether the model was able to remove the present serial correlation from the time series. All in all it gets obvious that modelling time series with AR properties is not a trivial task. Even though theoretical properties of AR processes are well researched, there are still some open questions when it comes to modelling such series.

A. Appendices

A.1. Appendix A. Code

GitHub: https://github.com/LHumpe/AR-1-Monte-Carlo-Simulation-fuer-TSA

A.1.1. Functions.R

```r
if (!require('matlib')) install.packages('matlib'); library('matlib')

#' Calculation of arithmetic mean
Mean <- function(y) {
  sumOfy <- sum(y)
  lenOfy <- length(y)
  return(sumOfy / lenOfy)
}

#' Calculation of empirical variance
Variance <- function(y){
  meanOfy <- Mean(y)
  lenOfy <- length(y)
  ySquared <- c()
  yCentered <- y - meanOfy

  for (t in 1:lenOfy){
    ySquared[t] <- yCentered[t] ^ 2
  }
  return((sum(ySquared) / (lenOfy - 1)))
}

#' Calculation of empirical autocovariance
Autocovariance <- function(y, maxlag = 10) {
  meanOfy <- Mean(y)
  lenOfy <- length(y)
  yCentered <- y - meanOfX
  autocovariances <- c()

  for (t in 0:maxlag) {
    autocovariances[t+1] <- sum(yCentered * lag(yCentered, -t)) / (lenOfy)
    }
  return(autocovariances)
}
```

```r
#' Calculation of the empirical autocorrelation
Autocorrelation <- function(y, maxlag = 10, plotting = TRUE) {
  meanOfy <- Mean(y)
  lenOfy <- length(y)
  yCentered <- y - meanOfy
  autocorrelations <- c()
  confidence = 1.96 / sqrt(lenOfy)

  for (t in 0:maxlag) {
    autocorrelations[t + 1] <- sum(yCentered * lag(yCentered, - t)) / (lenOfy)
  }

  autocorrelations <- autocorrelations / autocorrelations[1]

  if (plotting){
    PlotCorrelation(autocorrelations, confidence, maxlag, "ACF")
  }

  return(autocorrelations)
}

#' Calculation of partial autocorrelation using Yule-Walker Method
PartialAutocorrelation <- function(y, maxlag = 10, plotting = TRUE) {
  autocor <- Autocorrelation(y, maxlag = maxlag + 1, plotting=FALSE)
  autocorrelations = c(autocor[2])

  for (j in 2:maxlag){
    autocorrelations[k] <- (inv(toeplitz(autocor[1:j])) %*% autocor[2:(j+1)])[k]
  }

  if (plotting) {
    lenOfy <- length(y)
    confidence = 1.96 / sqrt(lenOfy)
    PlotCorrelation(partialAutocorrelations, confidence, maxlag, "PACF")
  }

  return(autocorrelations)
}
```

```r
#' Wrapper function for (P)ACF plots
PlotCorrelation <- function(corrObject, confidence, maxlag, ylabel) {
  yLower = min(-confidence, corrObject) - 0.005
  yUpper = max(confidence, corrObject) + 0.005

  if (ylabel == "ACF") {
    xMin = 0
  } else {
    xMin = 1
  }

  plot(corrObject,
       ylim = c(yLower, yUpper),
       xlim = c(xMin, maxlag),
       ylab = ylabel,
       xlab = "Lag",
       type = "n")

  abline(h = c(0, -confidence, confidence),
         col = c("black", "blue", "blue"),
         lty = c(1, 2 ,2))

  segments(xMin:maxlag, 0, xMin:maxlag, corrObject)
}

#' Simulates an autoregressive process
ar.sim <- function(n, mu, sigma, phi, padding = 100) {
  padding = padding
  n = n + padding

  epsilon <- rnorm(n, mean = mu, sd = sigma)
  y = c()
  y[1] <- epsilon[1]

  for (t in 2:n){
    y[t] <- y[(t - 1)] * phi + epsilon[t]
  }
  return(ts(x[(padding + 1):n]))
}
```

```
#' Monte-Carlo Simulation returning the first autocorrelation
mc.sim_pacf = function(n, phi, mu, sigma){
  phi.est=vector()

  for (i in 1:10000) {
    y = ar.sim(n, mu, sigma, phi)
    phi.est[i] = Autocorrelation(y, maxlag = 2, plotting = FALSE)[2]
  }

  se = sqrt(Variance(phi.est))

  return(list(mean(phi.est), mean(phi.est) - phi, se))
}
```

A.1.2. Process.R

```
source("Functions.R")

if (!require('ggplot2')) install.packages('ggplot2');
if (!require('doParallel')) install.packages('doParallel');
if (!require('gridExtra')) install.packages('gridExtra');
if (!require('kableExtra')) install.packages('kableExtra');

library('ggplot2')
library('doParallel')
library('gridExtra')
library('kableExtra')

no_cores <- 6
registerDoParallel(cores=no_cores)

set.seed(123)

mu = 0
sigma = 1
theorethical_phi = seq(from = -0.9, to = 0.9, by = 0.1)
N = c(25, 50, 100, 500, 1000)

result_frame = data.frame(phi = c(0), N = c(0),
                          phi_est = c(0), bias = c(0), se_phi = c(0))
```

```
i = 1
for (ac in theorethical_phi){
  sim_result <- foreach(i = N, .packages = 'matlib')
                          %dopar%
                          mc.sim_pacf(i, ac, mu, sigma)

  for (t in 1:length(N)){
    result_frame[i,] <- c(ac, N = N[t],
                          phi_est = sim_result[[t]][[1]] ,
                          bias = sim_result[[t]][[2]],
                          se_phi = sim_result[[t]][[3]])
    i <- i + 1
  }
}


ggplot(result_frame, aes(x = phi, y = se_phi, group = factor(N))) +
  geom_line() +
  geom_point(aes(shape = factor(N))) +
  scale_y_continuous(expression(SD ~ group("(",hat(phi),")"))) +
  scale_x_continuous(expression(phi),
                     breaks = theorethical_phi,
                     labels = as.character(theorethical_phi)) +
  scale_shape_discrete(labels = lapply(N, function(i) paste("T=", i, sep=" "))
  labs(shape="") +
  theme_minimal()


par(mfrow=c(1,2))

bias_plot <- ggplot(data = result_frame, aes(x = phi,
                                             y = bias,
                                             group = factor(N))) +
             geom_line() +
             geom_point(aes(shape = factor(N))) +
             scale_y_continuous(expression(Bias~of~hat(phi))) +
             scale_x_continuous(expression(phi),
                                breaks=theorethical_phi,
                                labels = as.character(theorethical_phi)) +
                                labs(shape = "") +
                                theme_minimal()
```

```
est_plot <- ggplot(result_frame, aes(x = phi,
                                     y = phi_est,
                                     group = factor(N))) +
            geom_line() +
            geom_point(aes(shape = factor(N))) +
            scale_y_continuous(expression(hat(phi)),
                               breaks = theorethical_phi,
                               labels = as.character(theorethical_phi)) +
            scale_x_continuous(expression(phi),
                               breaks = theorethical_phi,
                              labels = as.character(theorethical_phi)) +
            scale_shape_discrete(labels = lapply(N,
                                 function(i)paste("T=", i, sep=" "))) +
            labs(shape = "") +
            theme_minimal()

grid.arrange(est_plot, bias_plot, nrow = 2)

tbl_reshape <- reshape(result_frame, idvar = "phi",
                       timevar = "N", direction = "wide")

kable_column_names = c("Phi", rep(c("Phi_Est", "Bias", "Phi_Sd"), 5))

kable(tbl_reshape, digits = 3,
      row.names = F,
      col.names = kable_column_names,
      booktabs = T,
      escape = F,
      align = 'c') %>%
  add_header_above(c("","T=25"=3,"T=50"=3,"T=100"=3,"T=500"=3,"T=1000"=3)) %>%
  kable_styling(latex_options = "hold_position") %>%
  column_spec(1, bold = TRUE) %>%
  row_spec(0, bold = T) %>%
  collapse_rows(columns = 1)
```

A.2. Appendix B. Tables

	T=25			T=50			T=100			T=500			T=1000		
ϕ	$\hat{\phi}$	Bias	$SD(\hat{\phi})$	$\hat{\phi}$	Bias	$SD(\hat{\phi})$	$\hat{\phi}$	Bias	$SD(\hat{\phi})$	$\hat{\phi}$	Bias	$SD(\hat{\phi})$	$\hat{\phi}$	Bias	$SD(\hat{\phi})$
-0.9	-0.812	0.088	0.114	-0.852	0.048	0.076	-0.875	0.025	0.049	-0.895	0.005	0.020	-0.897	0.003	0.014
-0.8	-0.722	0.078	0.134	-0.758	0.042	0.093	-0.779	0.021	0.063	-0.796	0.004	0.027	-0.798	0.002	0.019
-0.7	-0.635	0.065	0.150	-0.667	0.033	0.104	-0.682	0.018	0.072	-0.696	0.004	0.032	-0.698	0.002	0.023
-0.6	-0.548	0.052	0.158	-0.574	0.026	0.113	-0.585	0.015	0.081	-0.597	0.003	0.035	-0.599	0.001	0.025
-0.5	-0.465	0.035	0.167	-0.481	0.019	0.120	-0.491	0.009	0.087	-0.498	0.002	0.039	-0.499	0.001	0.028
-0.4	-0.380	0.020	0.174	-0.388	0.012	0.127	-0.394	0.006	0.090	-0.398	0.002	0.041	-0.399	0.001	0.029
-0.3	-0.294	0.006	0.180	-0.296	0.004	0.130	-0.300	0.000	0.094	-0.300	0.000	0.042	-0.300	0.000	0.030
-0.2	-0.215	-0.015	0.181	-0.205	-0.005	0.136	-0.202	-0.002	0.097	-0.200	0.000	0.043	-0.201	-0.001	0.031
-0.1	-0.122	-0.022	0.187	-0.112	-0.012	0.136	-0.106	-0.006	0.099	-0.101	-0.001	0.044	-0.101	-0.001	0.032
0.0	-0.041	-0.041	0.187	-0.017	-0.017	0.137	-0.009	-0.009	0.099	-0.002	-0.002	0.045	-0.001	-0.001	0.032
0.1	0.046	-0.054	0.188	0.073	-0.027	0.137	0.084	-0.016	0.097	0.097	-0.003	0.044	0.098	-0.002	0.032
0.2	0.131	-0.069	0.186	0.165	-0.035	0.137	0.181	-0.019	0.097	0.197	-0.003	0.044	0.198	-0.002	0.031
0.3	0.215	-0.085	0.185	0.255	-0.045	0.132	0.279	-0.021	0.095	0.296	-0.004	0.043	0.298	-0.002	0.030
0.4	0.295	-0.105	0.185	0.347	-0.053	0.131	0.374	-0.026	0.092	0.395	-0.005	0.041	0.398	-0.002	0.029
0.5	0.381	-0.119	0.180	0.436	-0.064	0.127	0.468	-0.032	0.089	0.494	-0.006	0.038	0.497	-0.003	0.027
0.6	0.460	-0.140	0.174	0.531	-0.069	0.120	0.566	-0.034	0.083	0.593	-0.007	0.036	0.597	-0.003	0.025
0.7	0.539	-0.161	0.167	0.621	-0.079	0.112	0.661	-0.039	0.076	0.692	-0.008	0.032	0.696	-0.004	0.023
0.8	0.616	-0.184	0.160	0.710	-0.090	0.104	0.756	-0.044	0.068	0.791	-0.009	0.028	0.796	-0.004	0.019
0.9	0.684	-0.216	0.150	0.795	-0.105	0.092	0.849	-0.051	0.057	0.890	-0.010	0.021	0.895	-0.005	0.014

Table 1: Mean Estimates, Bias and Empirical Standard Error of r_1

B. References

Arnau, J. & Bono, R. (2001). Autocorrelation and bias in short time series: An alternative estimator. *Quality and Quantity*, *35*(4), 365–387.

Box, G. E. P. & Jenkins, G. . (1970). *Time series analysis : forecasting and control*. San Francisco, Holden-Day.

Brockwell, P. J. & Davis, R. A. (2016). *Introduction to Time Series and Forecasting* (3rd ed.). Cham, Springer International Publishing.

Huitema, B. E. & McKean, J. W. (1991). Autocorrelation Estimation and Inference With Small Samples. *Psychological Bulletin*, *110*(2), 291–304.

Krone, T., Albers, C. J. & Timmerman, M. E. (2017). A comparative simulation study of AR(1) estimators in short time series. *Quality and Quantity*, *51*(1).

Lawer, E. A. (2016). Empirical Modeling of Annual Fishery Landings. *Natural Resources*, *07*(04), 193–204.

McCleary, R., McDowall, D. & Bartos, B. (2017). Introduction, In *Design and analysis of time series experiments*. New York, Oxford University Press.

Scheffé, H. (1959). *The Analysis of Variance*. New York, Wiley.

Solanas, A., Manolov, R. & Sierra, V. (2010). Lag-one autocorrelation in short series: Estimation and hypotheses testing. *Psicologica*, *31*(2), 357–381.