# Report—SF2943 Time Series

Benjamin Kjellson

2014-05-07

# Problem 1

NB: Of all sections in this report, this may be the one that may seem "unnecessar[il]y lengthy". This is because what we are asked to do lead to combinatorially many things to investigate, in terms of tests to run, different sample size, processes and model coefficients, etc. Also note that the R functions that we will refer to later will be written in the syntax `package::function`, or possibly `package:::function` for hidden functions. Plots were made using the package ggplot2 [7].

## Objectives

Given a time series data set, we are to investigate whether it is likely that the data is a realization of white noise or even a realization of an iid sequence. Specifically, we aim to answer the following questions:

1. For a large enough sample (size $n$) from an iid sequence of random variables with finite variance, are the sample autocorrelations independent and distributed as $N(0, 1/n)$?
2. How effective are tests for the above question when we know the true process generating the data?
3. How do the tests described in section 1.6 of [1], also described below, perform on samples of different sizes from sequences of iid random variables or from low-order AR and MA processes with small coefficients?
4. For the latter type of processes, how large must the coefficients be in order for us to detect that they are not white noise sequences?
5. How does the sample size affect these tests?
6. How well do we detect a small trend in the data?

Since the process generating the data will be known to us, the perhaps most interesting part of our investigation will be to see how the different tests we use perform in comparison to each other. Below we describe the methodology and limitations of our implementation.

## Methodology and mathematical background

We will make 100 simulated samples for the iid data, and just as many for each low-order AR and MA process (i.e. for each set of model coefficients). Had we been dealing with single samples from each process, our first action would have been to plot the samples. With many samples however, we will instead aggregate the test results from each (type of) sample.

Specifically, we will run the tests described below, and look at the proportion of samples that do not reject null hypothesis (or equivalently the proportion that do reject the null hypothesis). Since we know the true data generating process, this will give us an indication of how prone each test is to give us Type I or Type II errors. To be consistent, we will consider a null hypothesis rejected if the $p$-value for the given test falls below the conventional value of 0.05.

### Normality and independence of the sample ACF

To investigate independence and normality of the sample autocorrelations, we can use the fact that we know[1] that our simulated sequence of iid random variables truly is iid, or vice versa, dependent for the AR and MA simulations. Beforehand, we decide to look at the first 40 sample autocorrelations, if only for the reason that this is the number mentioned in [1, p. 36]. Our smallest sample size is 50.

A point to note here is that the sample ACF (a) and portmanteau tests (b) mentioned in section 1.6 of [1] are for testing whether the sample itself is from an iid sequence, and not that the values of the sample ACF *from* such a sample are iid $N(0, 1/n)$, which is what we are interested in here. So while these tests will (should) fail to reject the null hypothesis of the sample autocorrelations being $N(0, 1/n)$ if this is actually the case, or reject the null hypothesis if, for example, the sample autocorrelations have a larger variance than $1/n$, it is not clear that these tests would reach the right conclusion about dependence or smaller variance than $1/n$ in the sample autocorrelations if that was the case. For example, it is conceivable that dependent observations of a particular distribution for the sample autocorrelations would produce a test statistic for the Ljung-Box test (described below) which is of smaller size than that needed to reject the null hypothesis. So in addition to conducting the two tests mentioned above, we shall also use tests specifically for distribution and independence in the same autocorrelations, as described next.

To test that the sample autocorrelations are independent and identically distributed as $N(0, 1/n)$ random variables, the following tests (abbreviations in parentheses) will be conducted:

- **Values of the sample ACF (sACF)**: As described in section 1.6 (a) of [1, p. 36], large samples (of size $n$) from an iid distribution with finite variance should have sample autocorrelations which are approximately iid $N(0, 1/n)$. If more than 5% (3 or more out of 40) of the sample autocorrelations fall outside the bounds $\pm 1.96/\sqrt{n}$, or if there is a clear outlier outside these bounds[2] then we reject the hypothesis that the sample autocorrelations are $N(0, 1/n)$ (though they could still be independent). The sample ACF up to $h$ lags can be calculated in R using the function `stats::acf`; this function calls a built-in C function which (presumably) uses the equations found in Definition 1.4.4 in [1, p. 19] for the calculations.
- Two other tests that rely on the iid $N(0, 1/n)$ assumption are the **Ljung-Box (LB)**[3] and **McLeod-Li (ML)** tests. If the sample autocorrelations are iid $N(0, 1/n)$, then the statistic $Q = n(n+2)\sum_{j=1}^{h} \hat{\rho}(j)/(n-j)$ will be approximately $\chi^2$ distributed with $h$ degrees of freedom. To maximize the power of the test (under the pretense that we do not know the true data generating distribution), we choose to use $h = \lfloor \log n \rfloor$ as per the recommendation in [2, p. 33]. The McLeod-Li test replaces the sample autocorrelations in the statistic above with the sample autocorrelations of the squared data, but is otherwise the same. The LB test is implemented in R in `stats::Box.test` and uses the `stats::acf` function to obtain the sample autocorrelations, and the function `stats::pchisq` to obtain a $p$-value from the $\chi^2(h)$ distribution for the $Q$ statistic calculated as above. For the ML test, this function is simply called with the squared data.

[1] To the extent that we trust the random number generator used

[2] We choose to define an outlier as one that is three standard deviations away from the mean. For a motivation of this heuristic, see http://en.wikipedia.org/wiki/Three_sigma_rule

[3] As implied in [1, p. 36] and virtually any other source on these tests, the Ljung-Box test is preferable to the Box-Pierce test, so we do not use it here.

- The **Turning point (TP) test** [1, pp. 36–37] is used to test the hypothesis that the data comes from a sequence of iid random variables. We will apply this to the sample autocorrelations. For three sequential observations, we say that there is a turning point if the middle one has a higher or lower value than the two others. For a large sample of size $n$, the number of turning points $T$ can be shown to be approximately normally distributed with mean $\mu_T = 2(n-2)/3$ and variance $\sigma_T^2 = (16n-29)/90$. We reject the null hypothesis if $\tau = (T-\mu_T)/\sigma_T$ falls outside the bounds $\pm 1.96$ (95% confidence interval for the mean of a standard normal distribution). The turning point test is implemented pretty much verbatim as described in [1, p. 37] by [5]. The function `turning.point.test` returns among other things a $p$-value associated with the above statistic $\tau$ using the cumulative distribution function of the normal distribution found in R as `stats::pnorm`.
- The **Kolmogorov-Smirnov (KS)** test is used to test that the sample autocorrelations are specifically from an $N(0, 1/n)$ distribution. The test statistic $D_n = \sup_x |F_n(x) - F(x)|$ measures the maximal difference in heigh between the empirical distribution function $F_n$ and the reference distribution $F$ [3], which in our case is the $N(0, 1/n)$ distribution. Appropriate quantiles are provided in any statistical package that supports the test. We choose the KS test and not the Shapiro and Francia or Jarque-Bera tests described in [1, p. 38] because we want to test for a *specific* normal distribution, and not normality in general. The KS test is implemented in R in `stats::ks.test`, and returns among other things a $p$-value we may use for the above purpose.

When we conduct these tests for data coming from a sequence of random variables known to be iid (i.e. the simulated iid data), we obtain the proportion (among the 100 simulations) of errors we make in rejecting the null hypothesis, i.e the proportion of type I errors. If the tests work as they should, this proportion should match our significance level, 0.05. When we conduct these tests for data from a dependent sequence, the proportion of null rejections will indicate the power of the test (and one minus this number the proportion of type II errors). Together, these two situations answer the question of effectiveness (producing the desired result).

### Other tests: (in)dependence and trends

After conducting the above investigation, if we now believe that the sample autocorrelations from independent sequences are indeed iid $N(0, 1/n)$, and are satisfied with the performance of the above tests that rely on this assumption, we may use these tests to identify whether a sequence of random variables is iid or not. We therefore apply the turning point test again to the full samples (and not just the sample autocorrelations).

The **difference-sign (DS) test** and the **rank test (RT)**, as described in [1, p. 37], also operates with a null hypothesis that the data are iid, and are useful for detecting a trend in the data. The DS test counts the number $S$ of times that a value in the sample exceeds the previous one. For an iid sequence of length $n$, that number has an expected value of $\mu_S = (n-1)/2$ and a variance of $\sigma_S^2 = (n+1)/12$, and for large $n$, $S$ will be distributed as $N(\mu_S, \sigma_S^2)$. A test statistic and test may then be formulated analogously to the TP test. The rank test counts the number $P$ of times each data point in the sample is exceeded by a later data point. $P$ has an expected value of $\mu_P = n(n-1)/4$ and a variance of $\sigma_P^2 = n(n-1)(2n+5)/72$ if the sequence is iid, and for large $n$, $P \sim N(\mu_P, \sigma_P^2)$. Again the test statistic and test is analogous to the TP test. These tests are also found in [5] by the names `difference.sign.test` and `rank.test` and are implemented just

as described in [1, p. 37], with return structure similar to the implementation of the TP test. We will try these two tests on the iid data (noise), the iid data with a linear trend with slope ranging between 0.01 and 1 (1 being the standard deviation of the noise), and the AR and MA sequences.

To test the less stringent condition that the data is white noise, one can, as described in [1, p. 38], fit an AR model to the data using the Yule-Walker equations and choose the order which minimizes the AICC statistic (see definition below). The problematic part here is that the YW equations do not necessarily maximize the true likelihood, but the AIC, and by extension the AICC, are *defined* through the maximized likelihood in the way they are constructed. In R, specifically for the function `stats::ar`, the workaround for this issue is to instead calculate the information criterion by using the YW estimate of the variance instead of the maximum likelihood estimate, and omitting the determinant term from the (Gaussian) likelihood [4]. Note that the R function `ar.yw` finds the best order by minimizing the AIC; I wrote the following function which instead finds the best order according to the AICC:

```
# stats::ar.yw gives
# "The differences in AIC between each model and the best-fitting model"
# Need only to add the additional term in the AICC and find minimum
# to get the optimal order according to the AICC.
ar.yw.order.aicc <- function(x, order.max=10) {
  n <- length(x)
  k <- 0:order.max
  aic <- ar.yw(x, aic=TRUE, order.max=order.max)$aic
  aicc <- aic +  2 * k * (k + 1) / (n - k - 1)
  return(unname(which.min(aicc) - 1))
}
```

Finally, since the noise processes for our simulated AR and MA models will be Gaussian[4] we can check that the fitted models return residuals that appear to be so as well. For $n \leq 200$ we use the Shapiro-Francia test of normality described in [1, p. 38] and implemented in R in `nortest:sf.test` (which uses a slighly different implementation than that described in the course book). For $n > 200$, we use the Jarque-Bera test as described in section 5.3, [1, p. 167], and implemented in R as `tseries::jarque.bera.test`.

The AIC statistic for a model with parameter vector $\boldsymbol{\beta}$ of size $k$ and likelihood $L$ is defined as $\mathrm{AIC}(\boldsymbol{\beta}) = -2 \log L(\boldsymbol{\beta}) + 2k$. A smaller AIC indicates a better model all else equal; it can be seen that the term $2k$ penalizes models with too many parameters. The AICC imposes a larger penalty term, being defined as $\mathrm{AICC}(\boldsymbol{\beta}) = \mathrm{AIC}(\boldsymbol{\beta}) + 2k(k+1)/(n-k-1)$. These definitions were taken from [6], and are equivalent to those found in [1, p. 173]

[4] Chosen for simplicity and since Problem 1 is long enough as it is.

## Results

We first test the claim that the sample autocorrelations are iid $N(0, 1/n$ when the sequence generating the sample is itself iid with finite variance. We simulate 100 samples of sizes 50, 100, 500, 1000 and use the tests described above to investigate this claim. To reiterate, these tests are the sACF test (to abbreviate the procedure described), the Ljung-Box (LB) and McLeod-Li (ML) tests, the turning-point (TP) test, and the Kolmogorov-Smirnov (KS) test, using the parameters mentioned previously. The null hypotheses for these tests are aligned with the claim made, so that we hope to see a large fraction of the tests **not** rejecting the null hypothesis. The results are shown in figure 1 below.

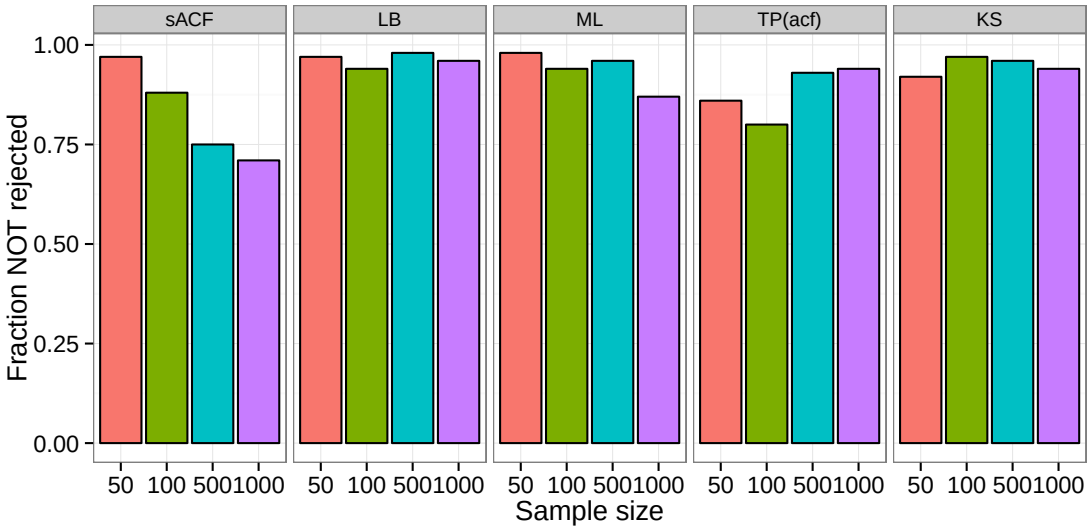


Figure 1: Tests for the hypothesis that sample autocorrelations are independent and normally distributed, with simulated iid data. y-axis show proportion of tests that did NOT reject the null hypothesis

We see that the LB, ML and KS tests perform fairly well here, seemingly independent of sample size. The performance of the sACF test seems to decrease with sample size, while the TP test does the opposite. Overall, these tests point to the conclusion that sample autocorrelations from iid sequences with finite variance are indeed approximately normally distributed with mean zero and variance $1/n$ for $n$ large. Since we know this is true theoretically, the more interesting conclusion is that we now have an indication of which tests work well when the null hypotheses for these tests are true.

Let us therefore look at the situation when the null hypotheses are not true, by looking at simulated sequences from AR(1) and MA(1) with coefficients ranging between $-0.9$ and 0.9 (0 not included). Using the same setup as above, figure 2 shows the test results from the above types of dependent sequences:

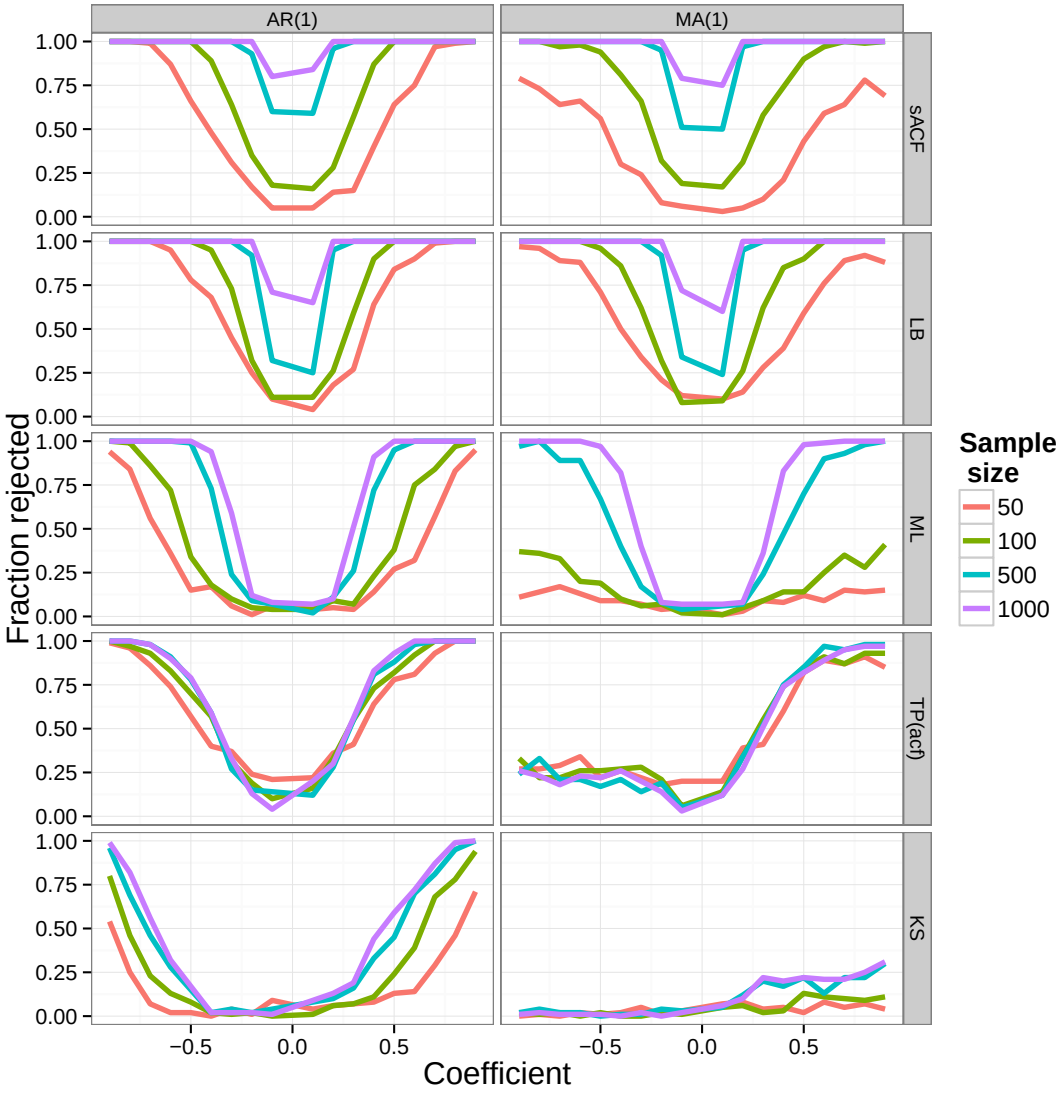


Figure 2: Tests for the hypothesis that sample autocorrelations are independent and normally distributed, with simulated dependent data. y-axis show proportion of tests that did NOT reject the null hypothesis

The figure shows the fraction of tests in which the null hypothesis was rejected. Since the null hypotheses ought to be false in this situation, the figure gives us an indication of the power of these tests. Particularly, the KS and TP tests do not fare well for MA processes, though the TP test understandably does better and better as the MA coefficient becomes larger (more positive). Comparatively, the sACF, LB and ML tests do much better when the coefficients are small, and power increases with sample size. It can also be seen that for small sample sizes, the ML test has much less power than the LB test.

Turning to the ability of detecting a trend in the data, we add a linear trend with different slopes (0, 0, 0.01, 0.1, 0.25, 0.5, 1) to the simulated iid sequences, and look at what proportion of each test (difference-sign and rank test) reject the null hypothesis of the data being iid (no trend). The results are shown in figure 3:

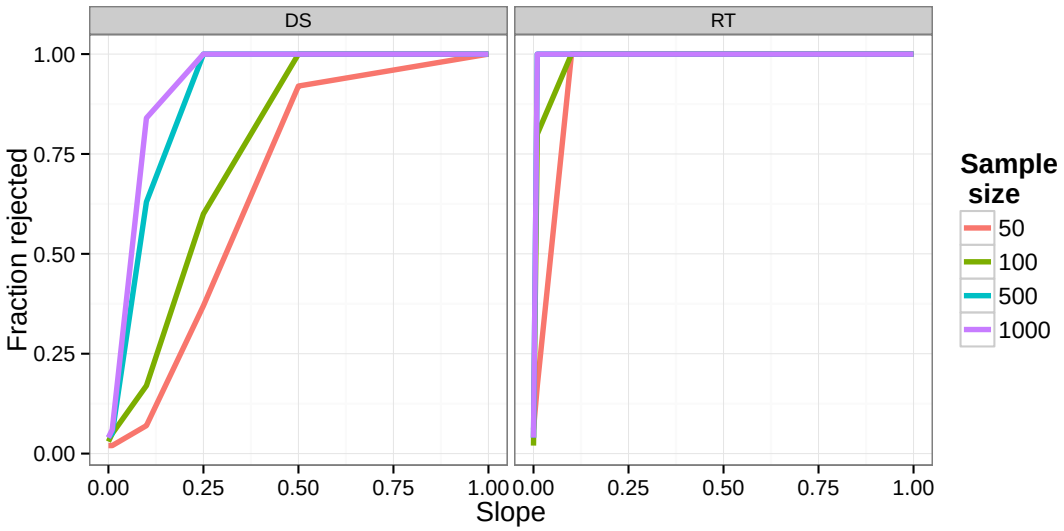


Figure 3: Proportion of tests (difference-sign (DS) and rank test (RT) that reject the assumption of no trend, for simulated iid data with a linear trend added.

As can be seen, the rank test performs better than the difference-sign test in identifying a trend in the data when there is one, with increasing sample size improving the power of each test.

Next we run the remaining tests on the iid standard normal data, looking to see how effectively the turning point test identifies the data as iid; if an AR model fitted to the data with order selected by minimizing the AICC indeed yields an order of zero, and lastly if the data is identified as normal by the Francia-Shapiro or Jarque-Bera tests. The results are shown in figure 4.

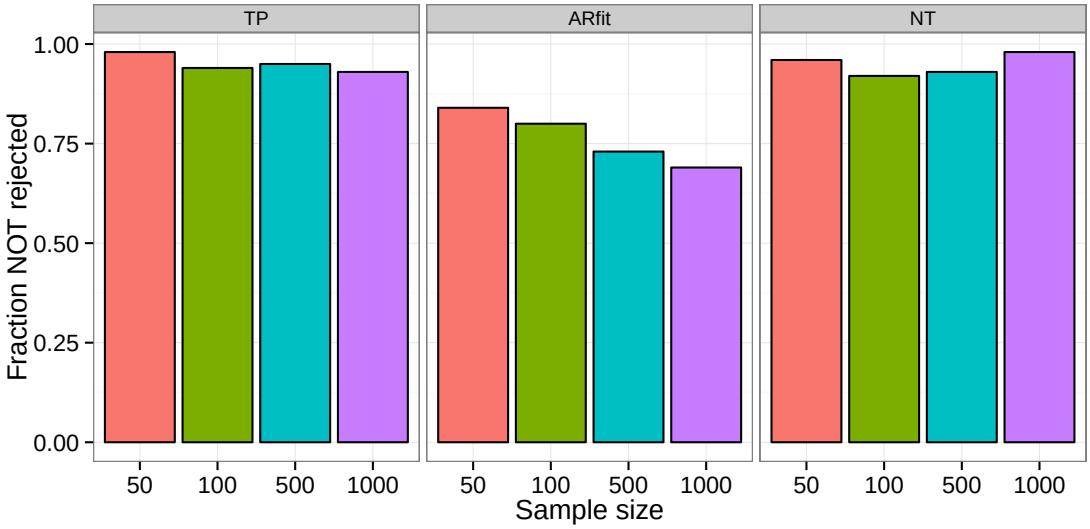


Figure 4: Proportion of tests that lead to the correct conclusion about the data, for data simulated from a standard normal distribution.

As can be seen, the turning-point test does a fairly good job of identifying iid data as such overall, with no discernable pattern in effectiveness as the sample size increases. The tests for normality produce fairly good results as well, though there is a slight decrease in performance as the sample size increases. Lastly, the test for white noise, consisting of finding the optimal order of an AR model by minimizing the AICC, does fairly poorly in identifying the correct order of zero, but it is hard to see any patterns of performance in the sample size.

We likewise run the remaining tests for the AR and MA processes, the results shown in figure 5 and interpreted below.

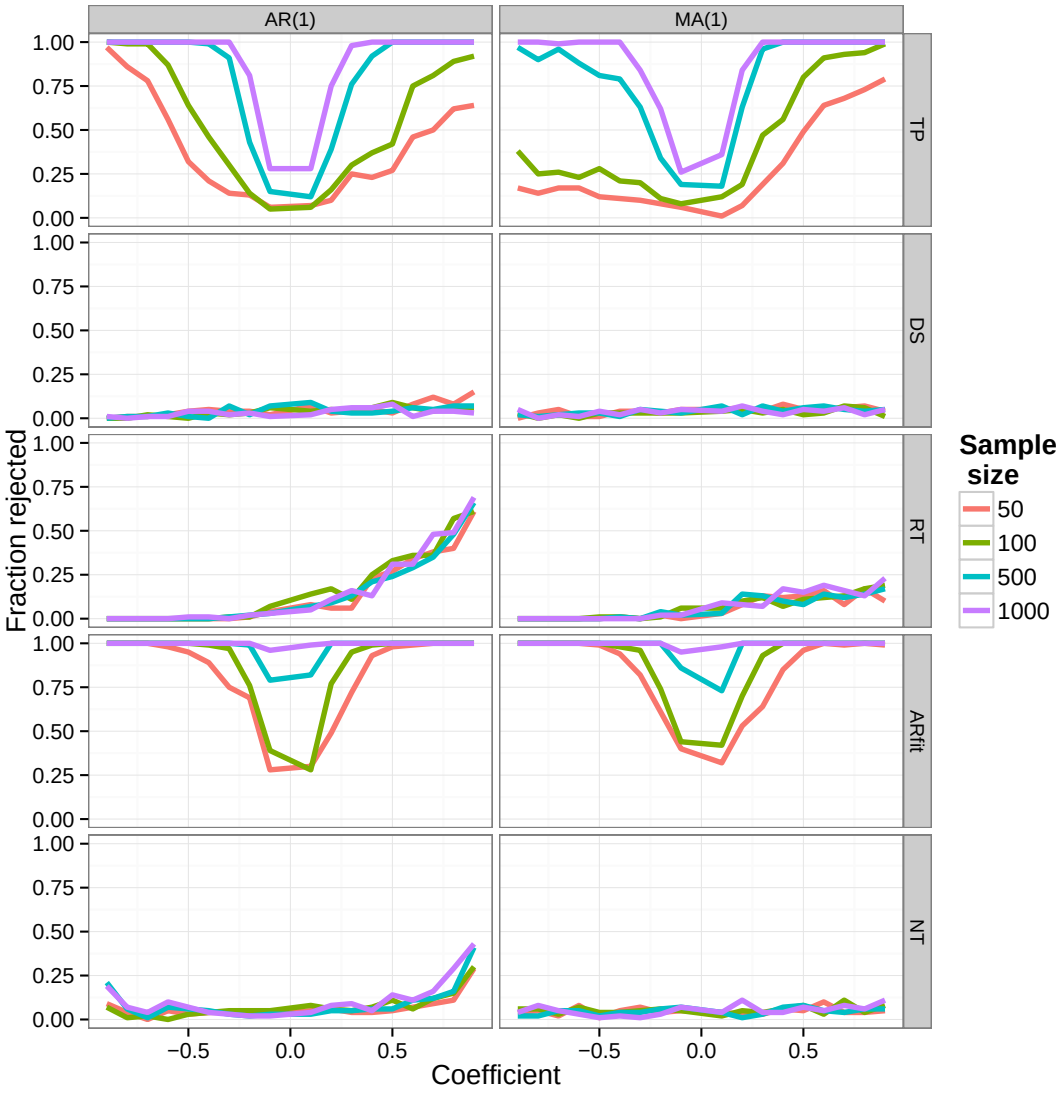


Figure 5: Proportion of tests that lead to the correct conclusion about the data, for data simulated from AR(1) and MA(1) processes.

We see that the turning point test does fairly well in rejecting the null hypothesis of iid data, with performance increasing with sample size. The difference-sign and rank tests, which also rely on an assumption of iid data, do much worse, though the rank test shows slight increase in power as the size of the AR(1) or MA(1) coefficient increases (in the positive direction). Fitting an AR model by minimizing the AICC seems to do a good job of selecting an order higher than zero for both the AR(1) and MA(1) processes. The latter type of process can be represented as infinite-order AR processes and as such should at least have its order identified as non-zero. We also see that the sample size increases the power for this white noise test. The normality tests, lastly, fail to reject the null hypothesis of a normal distribution, though it is unclear how close to a normal distribution AR and MA processes with normally distributed noise (and starting values/burn-in) really is.

## Summary

By simulation, we have established that the sample autocorrelations from an iid sequence with finite variance indeed seem to have a $N(0, 1/n)$ distribution as claimed. The Ljung-Box test seemed to perform best in this aspect, in terms of correctly identifying whether or not the data came from such a sequence or not, having higher power than the McLeod-Li test. The rank test was found to be clearly better than the difference-sign test at identifying linear trends in the data, while the turning point test can be used with some success in identifying data as iid or not. Testing for white noise by fitting several AR models to the data and choosing the one which minimized the AICC worked fairly well in identifying an order higher than zero when that was really the case for the data generating process. However, to a certain extent it also identified a non-zero order when the data generated by an iid process, suggesting at the very least that this method has its flaws.

# Problem 2

For this problem we are considering the AR(2) time series model

$$X_t - 1.3X_{t-1} + 0.65X_{t-2} = Z_t, \quad \{Z_t\} \sim \text{WN}(0, 280) \tag{1}$$

where the white noise sequence consists of independent and normally distributed random variables.

## Objectives

Our objectives are, with respect to model (1), to

- Determine if the time series is stationary
- Determine if the time series is causal
- Compute and plot the autocorrelation function
- Compute and plot the spectral density
- Explain how these two plots belong to model (1)

## Mathematical Background

Consider first the geneneral AR(2) process

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} = Z_t, \quad Z_t \sim \text{WN}(0, \sigma^2) \tag{0.1}$$

The associated autoregressive polynomial is

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 \tag{0.2}$$

and it has roots

$$z_1 = -\frac{\phi_1}{2\phi_2} + \sqrt{\frac{\phi_1^2}{4\phi_2^2} + \frac{1}{\phi_2}} \tag{0.3}$$

$$z_2 = -\frac{\phi_1}{2\phi_2} - \sqrt{\frac{\phi_1^2}{4\phi_2^2} + \frac{1}{\phi_2}} \tag{0.4}$$

If $|z_1| \neq 1$ and $|z_2| \neq 1$, then (0.1) is stationary; if these roots lie outside the unit disk, then the process is also causal. Below, assume that this is so.

If we write $X_t = \phi_1 X_{t-1} + X_{t-2} + Z_t$, multiply by $X_{t-h}$ on both sides and take expectations, we get

$$\begin{aligned} \text{E}[X_t X_{t-h}] &= \phi_1 \text{E}[X_{t-1} X_{t-h}] + \phi_2 \text{E}[X_{t-2} X_{t-h}] + 0 \quad \Leftrightarrow \\ \gamma(h) &= \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2) \quad \Leftrightarrow \\ \rho(h) &= \phi_1 \rho(h-1) + \phi_2 \rho(h-2) \end{aligned}$$

upon division by $\gamma(0)$. Noting that $\rho(0) = 1$ we set $h = 1$ and use the symmetry of the autocorrelation function with respect to $h$ to get $\rho(1) = \phi_1 \rho(0) + \phi_2 \rho(1)$, which we may solve as

$$\rho(1) = \frac{\phi_1}{1 - \phi_2} \tag{0.5}$$

Autocorrelations for $h > 1$ may then be obtained recursively through the equation

$$\rho(h) = \phi_1 \rho(h-1) + \phi_2 \rho(h-2) \tag{0.6}$$

The spectral density of the AR(2) process is

$$f(\lambda) = \frac{\sigma^2}{2\pi(1 + \phi_1^2 + 2\phi_2 + \phi_2^2 + 2(\phi_1\phi_2 - \phi_1)\cos\lambda - 4\phi_2\cos^2\lambda)}, \tag{0.7}$$

defined for $-\pi \leq \lambda \leq \pi$ [1, p. 133]. Restricting $\lambda$ to the interval $[0, \pi]$, the $\lambda$ that maximizes $f$ is seen to be

$$\lambda^* = \arccos\left(\frac{\phi_1\phi_2 - \phi_1}{4\phi_2}\right) \tag{0.8}$$

## Results

Now we apply the above to the model (1). For this model, $\phi_1 = 1.3$ and $\phi_2 = -0.65$, and the autoregressive polynomial is seen to have roots $z = 1 \pm \sqrt{\frac{7}{13}}i$, so that $|z| = \sqrt{\frac{20}{13}} > 1$. The given process is thus stationary and causal. Using equation (0.5) and (0.6), its ACF is defined by $\rho(0) = 1$, $\rho(1) = \frac{26}{33} \approx 0.7879$ and $\rho(h) = 1.3\rho(h-1) - 0.65\rho(h-2)$, which may be used to plot the autocorrelation function for further lags as in figure 6:

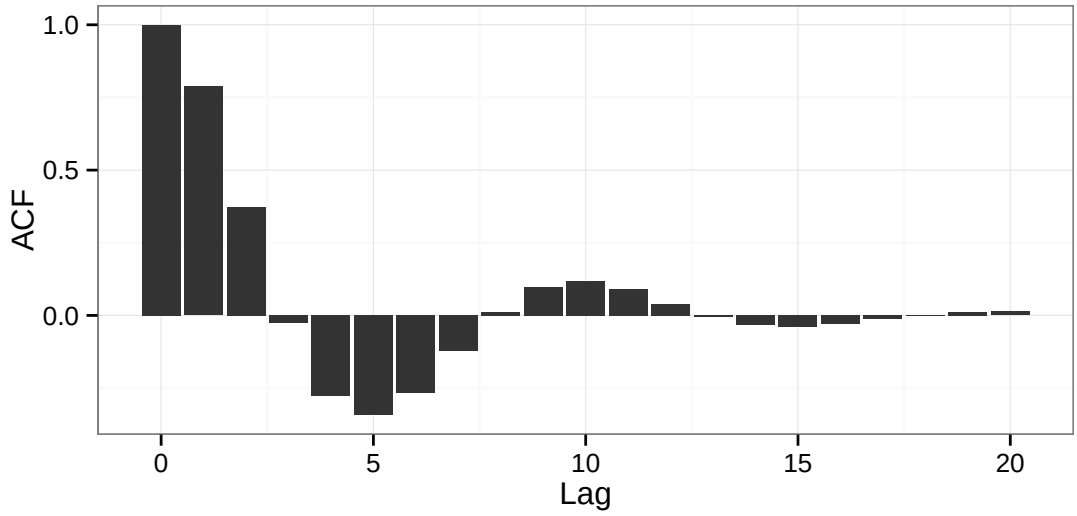


Figure 6: Autocorrelation function for model (1).

Similarly, inserting the coefficients of model (1) into equation (0.7) and plotting the result yields

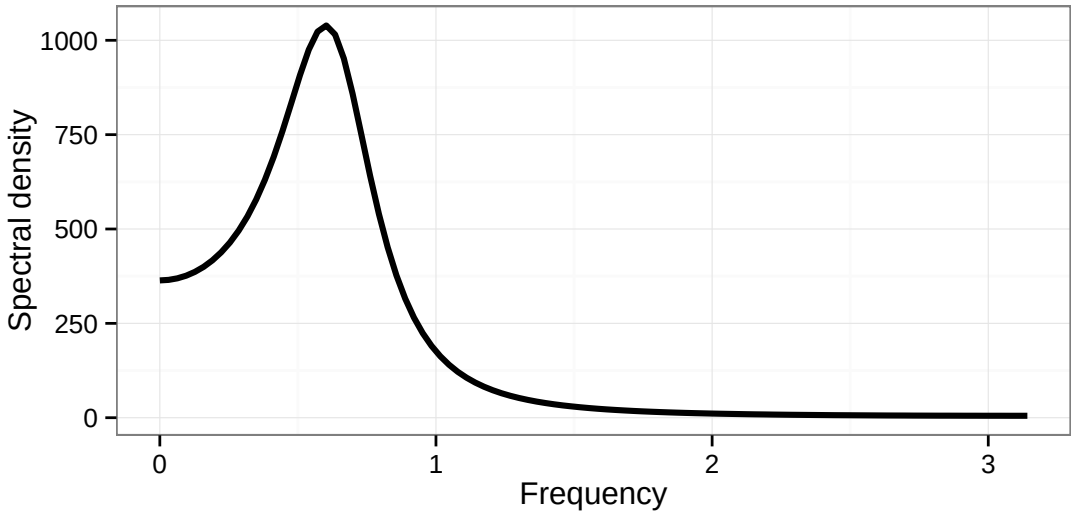


Figure 7: Spectral density for model (1).

with a maximum at $\lambda^* = \arccos(33/40) \approx 0.6$ and a maximum value of approximately 1039, which seems to correspond to what is seen in the plot.

As for the question of how these plots can be seen to belong to model (1), knowledge of the model and the specific values of its coefficients, along with the formulas for the autocorrelation function and spectral density above, can be used to make this identification. As stated, the theoretical coordinate of the maximum, which is approximately (0.6, 1039), of the spectral density function is seen to be matched in the plot. Likewise, the lag 1 autocorrelation of 0.7879 can be seen to match the lag 1 correlation of the plot, and if there is still uncertainty, more of the theoretical autocorrelations could be calculated from (0.6). The situation would be much more difficult if we only had access to a sample from the process, as there is no guarantee that the sample equivalents of the autocorrelation function and the spectral density would look anything like the true versions.

## Summary

We have made a theoretical investigation about the autocorrelation function and spectral density for a causal AR(2) process, finding recursive or closed form solutions for these functions. We looked at a specific example of such a process in model (1).

# Problem 3

## Objectives

We are to simulate 100 samples of size 200 from the AR(2) model (1), with $\{Z_t\}$ coming from a $N(0, 280)$ distribution. Then we are to

- Estimate the model parameters for each sample using two different methods, and make a scatterplot of the estimated model coefficients.
- Compute the (empirical) one-step mean square prediction error using both methods, to find which model performs best (produces the smallest error).
- Make a histogram of these errors.
- Investigate whether the filtered residuals from the fitted model and the simulated data have the same (normal) distribution as the original simulated sample $\{Z_t\}$.
- Also fit an AR(10) model to the data, and compare its 1- and 3-step prediction errors to one of the AR(2) models fitted previously. Here we choose the Burg method for model fitting for the AR(2) and the AR(10) models.

## Methodology and Mathematical Background

The first method for estimating the parameters of model (1) is **Yule-Walker estimation**. [1, pp. 139–143] go into greater detail on this method, but we summarize the parts most needed for our purpose here. A causal AR($p$) process $\{X_t\}$ with autoregressive polynomial $\phi(z)$ may be represented as $X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}$, with $\psi_j$ being the coefficient of $z^j$ in the power series $1/\phi(z)$ and $\{Z_t\} \sim WN(0, \sigma^2)$ being the noise/innovations of the process. By multiplying both sides of the previous equation by $X_{t-j}$ for $j = 0, 1, \ldots, p$ and taking expectations, the **Yule-Walker equations** are obtained as

$$\Gamma_p \boldsymbol{\phi} = \boldsymbol{\gamma}_p, \quad \text{and} \tag{0.9}$$

$$\sigma^2 = \gamma(0) - \boldsymbol{\phi}' \boldsymbol{\gamma}_p \tag{0.10}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)'$, $\Gamma_p$ is the covariance matrix $[\gamma(i-j)]_{i,j=1}^{p}$, and $\boldsymbol{\gamma}_p = (\gamma(1), \ldots, \gamma(p))'$. Knowing $\boldsymbol{\phi}$ and $\sigma$, these equations can be used to solve for $\gamma_p$.

In practice, $\phi$ and $\sigma$ are the quantities we want to estimate. Letting the sample estimates of the above symbols be denoted by adding a caret ($\hat{}$) to the previous symbol, and assuming $\hat{\gamma}(0) > 0$, we can define $\hat{R}_p = \hat{\Gamma}_p / \hat{\gamma}(0)$ and $\hat{\boldsymbol{\rho}}_p = \hat{\boldsymbol{\gamma}}_p / \hat{\gamma}(0) = (\hat{\rho}(1), \ldots, \hat{\rho}(p))'$. With these estimates, the **sample Yule-Walker equations** are given by

$$\hat{\boldsymbol{\phi}} = (\hat{\phi}_1, \ldots, \hat{\phi}_p)' = \hat{R}_p^{-1} \hat{\boldsymbol{\rho}}_p, \quad \text{and} \tag{0.11}$$

$$\hat{\sigma}^2 = \hat{\gamma}(0) \left[ 1 - \hat{\boldsymbol{\rho}}_p' \hat{R}_p^{-1} \hat{\boldsymbol{\rho}}_p \right] \tag{0.12}$$

These are implemented in R in the function `stats::ar.yw`, which returns among other things the estimated coefficients (`$ar`) and the estimated noise variance (`$var.pred`) (which really is calculated according to the above equation according to the function documentation).

The second method we use to estimate the parameters of model (1) is using **Burg's algorithm**. Again we draw from [1, pp. 147–148] but only cover the essential parts for our purpose. This method "estimates the PACF $\{\phi_{11}, \phi_{22}, \ldots\}$ by successively minimizing sums of squares of forward and backward one-step prediction errors with respect to the coefficients $\phi_{ii}$". For a sample $\{x_1, \ldots, x_n\}$ of a zero-mean stationary process $\{X_t\}$, the forward one-step prediction error $u_i(t)$ for $t = i+1, \ldots, n$ is defined as the difference between $x_{n+1+i-t}$ and its best linear estimator in terms of the previous $i$ observations, and the backward one-step prediction error $v_i(t)$ is defined similarly for $x_{n+1-t}$ in terms of the subsequent observations. With $u_0(t) = v_0(t) = x_{n+1-t}$, **Burg's algorithm** consists of solving the following recursions for $i = 1, \ldots, p$:

$$d(1) = \sum_t = 2^n \left( u_0^2(t-1) + v_0^2(t-1) \right), \tag{0.13}$$

$$\phi_{ii} = \frac{2}{d(i)} \sum_{t=i+1}^{n} v_{i-1}(t) u_{i-1}(t-1), \tag{0.14}$$

$$d(i+1) = \left(1 - \phi_{ii}^2\right) d(i) - v_i^2(i+1) - u_i^2(n), \tag{0.15}$$

$$\sigma_i^2 = \left(1 - \phi_{ii}^2\right) d(i)/2(n-i) \tag{0.16}$$

This algorithm is implemented in R in the function `stats::ar.burg`, with return structure similar to that of `stats::ar.yw`.

The models fitted by either the Yule-Walker equations or by Burg's algorithm will be causal and will both have approximately the same large-sample distribution as the maximum likelihood estimates of the model parameters [1, pp. 140,148], though Burg's algorithm usually gives higher likelihoods for AR($p$) models [1, p. 139].

The one-step prediction of an AR($p$) model with mean $\mu$ at time $n$, $n \geq p$, conditional on $X_t = x_t$, $t = 1, \ldots, n$, is given by [1, p. 68]

$$P_n X_{n+1} = \mu + \phi_1 x_n + \ldots + \phi_p x_{n+1-p} \tag{0.17}$$

The prediction $P_n X_{n+h}$ can be obtained recursively by (see [1, p. 104] or [2, p. 56])

$$P_n X_{n+h} = \mu + \sum_{i=1}^{p} \phi_i P_n X_{n+h-i}, \tag{0.18}$$

where it is understood that $P_n X_{n+h-i}$ is replaced by $X_{n+h-i} = x_{n+h-i}$ if $h - i \leq n$. In practice, the mean $\mu$ and the coefficients $\phi_i$ are replaced by their sample estimates $\hat{\mu}$ and $\hat{\phi}_i$. This prediction methodology is the one used by the function `stats:::predict.ar` in R, which we will use below.

The mean square ($h$-step) prediction error, MSPE($h$), is defined as

$$\text{MSPE}(h) = \text{E}\left[ (X_{n+h} - P_n X_{n+h})^2 \right] \tag{0.19}$$

The processes we are considering are stationary with white noise innovations (i.e. finite variance), so if we are able to simulate from the process we want to determine the MSPE of, we can use the law of large numbers to estimate the MSPE by

$$\widehat{\text{MSPE}}(h) = \frac{1}{M} \sum_{j}^{M} (X_{n+h}^{(j)} - P_n X_{n+h}^{(j)})^2 \tag{0.20}$$

where the superscript $(j)$ denotes the $j$th simulated sample $\{x_1^{(j)}, \ldots, x_n^{(j)}, x_{n+1}^{(j)}, \ldots, x_{n+h}^{(j)}\}$, $j = 1, \ldots, M$, and model fitting is only done on the data $\{x_1^{(j)}, \ldots, x_n^{(j)}\}$.

## Results

We simulate 100 samples of size 100 from the AR(2) model (1), with $\{Z_t\}$ distributed as $N(0, 280)$. To these samples, we fit AR(2) models using the Yule-Walker equations and Burg's algorithm with the R functions mentioned above. In figure 8 we see a scatterplot of the estimated model coefficients. As can be seen, the estimates seem to be negatively correlated. This is expected as a larger $\hat{\phi}_1$ would increase the importance of the most recent data point, and $\hat{\phi}_2$ would therefore have to become more negative in order for the second-to-last data point to compensate.

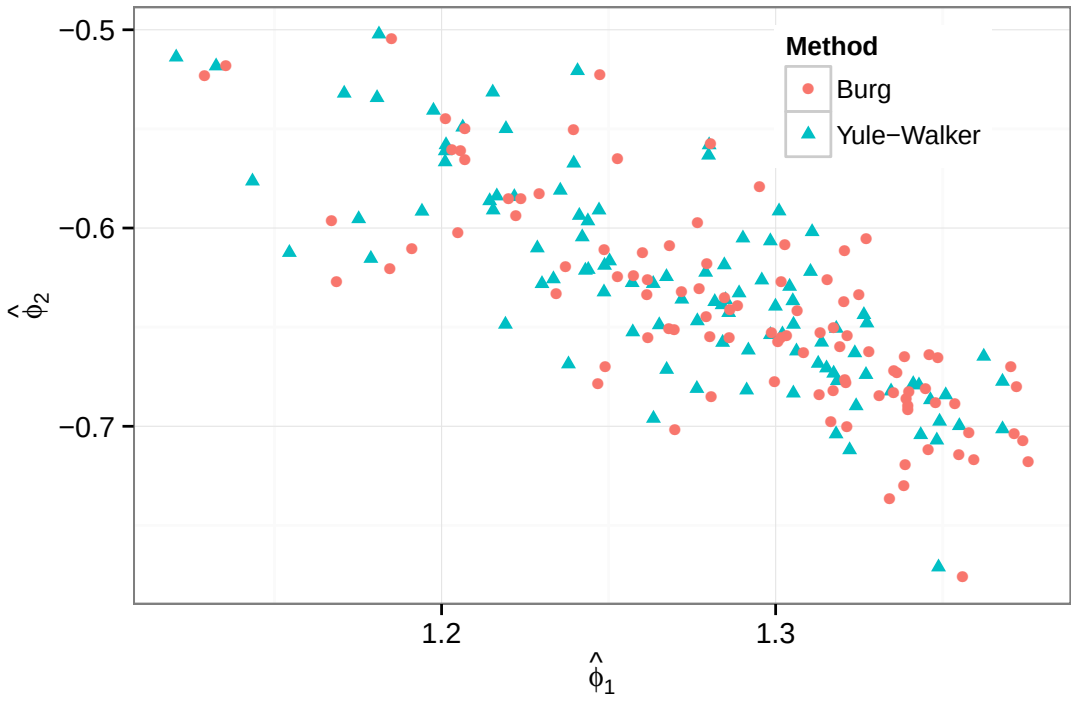


Figure 8: Estimated coefficients for model (1) by the Yule-Walker equations and by Burg's algorithm.

We also see from this plot that there seems to be a slight tendency for the Yule-Walker equations to favor smaller values of the coefficients, but why this should be so is not immediately clear.

The estimated one-step MSPE is 264.5 for the model fitted by the Yule-Walker equations and 263.8 for the model fitted by Burg's algorithm; the latter estimation works slightly better in this regard. As seen in figure 9, there seems to be no substantive difference in the prediction errors between the two models.

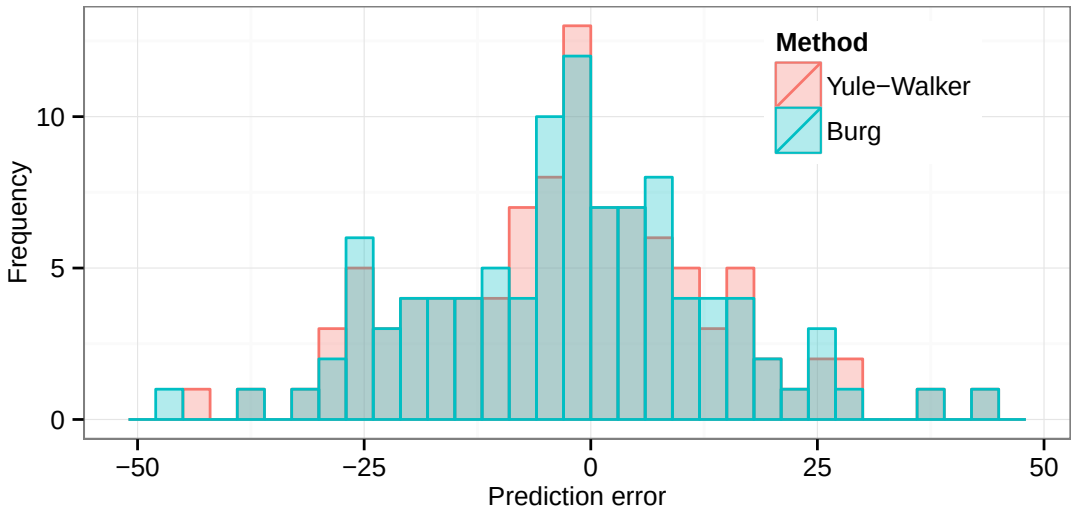


Figure 9: One-step prediction errors delivered by models fitted by the Yule-Walker equations and by Burg's algorithm.

As for the distribution of the residuals for the fitted models, and the simulated noise we used in creating the samples from model (1), we ran the sACF and Kolmogorov-Smirnov tests discussed in Problem 1, and looked at the proportion of tests out of the 100 samples that rejected or failed to reject the hypothesis that these residuals come from a $N(0, 280)$ distribution. The simulated noise correctly failed to reject the null hypothesis 82% of the time for the sACF test and 92% of the time for the KS test, giving some indication that the random number generator works as it should. For the residuals from the Yule-Walker model, the corresponding percentages are 83% and 99%, surprisingly better than the noise simulated directly from the distribution of the null hypothesis. However, as we saw in problem 1 these tests have low power to reject a false null hypothesis. Finally, the percentages for the tests of the Burg model residuals are 82% and 100%; no substantive difference in comparison to the YW model.

We also fitted an AR(10) model to the data using Burg's algorithm, and will now compare its one- and three-step prediction errors to the AR(2) model fitted with the same method. Figure 10 shows the one-step prediction errors for the fitted AR(2) and AR(10) models. As can be seen, the distributions of the prediction errors seem quite similar, though perhaps with slightly wider spread for the AR(10) model. As stated in [1, p. 169], a higher MSPE is to be expected for models with more parameters, since the MSPE will depend not only on the white noise variance of the fitted model (which on the other hand is smaller for models with more parameters), but also on the parameter estimates—and there are more of them.

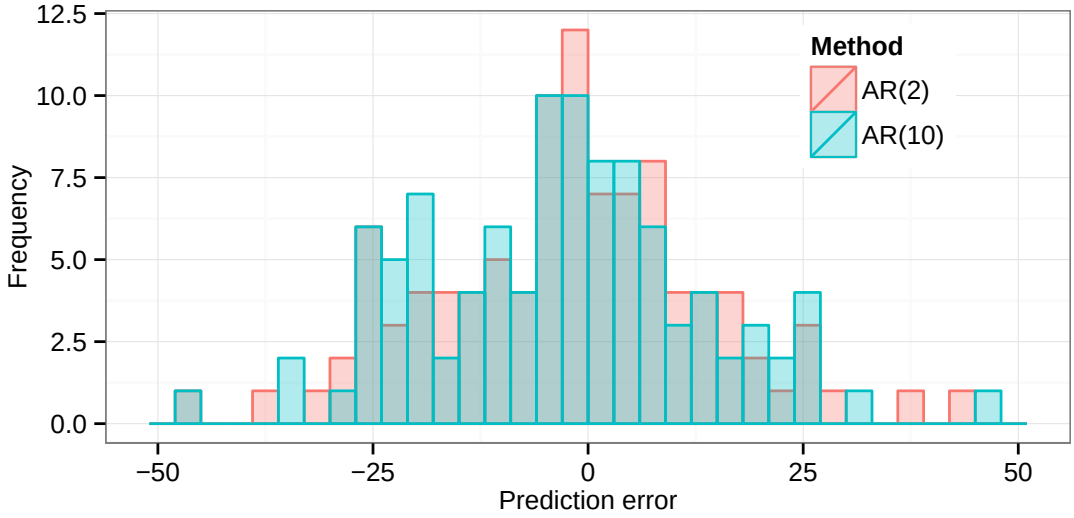


Figure 10: One-step prediction errors delivered by AR(2) and AR(10) models fitted by Burg's algorithm.

The estimated one-step MSPE for the AR(2) model is as mentioned above 263.8; while for the AR(10) model it is 268.4. As expected, the MSPE is higher for the AR(10) models.

Three-step prediction errors were also calculated for the fitted AR(2) and AR(10) models, and are shown in 11. The errors are again not too different in how they are distributed.

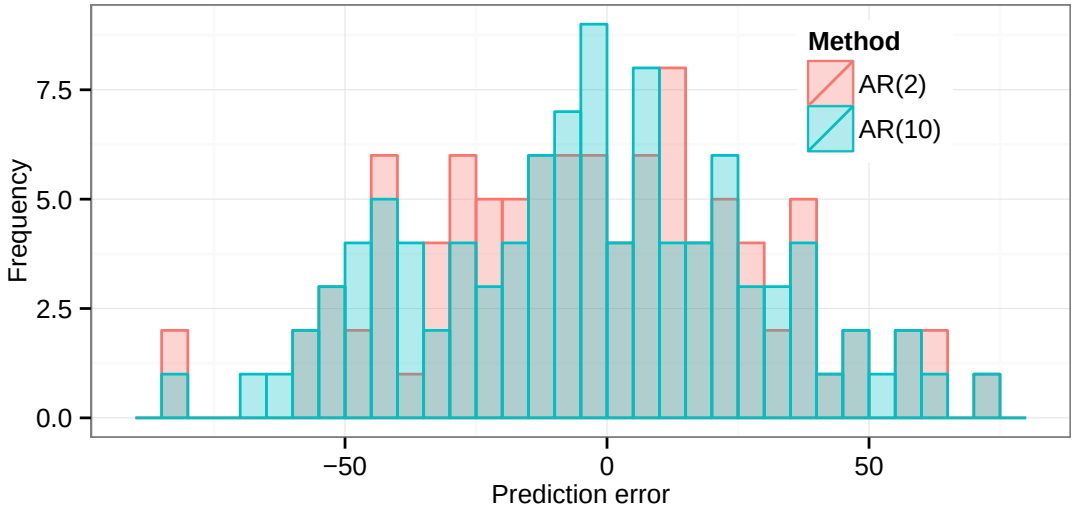


Figure 11: Three-step prediction errors delivered by AR(2) and AR(10) models fitted by Burg's algorithm.

The estimated three-step MSPE for the AR(2) model is 1012.4; while for the AR(10) model it is 1046.1. As expected, the MSPE is higher for the AR(10) models.

### Summary

We have tried and compared the Yule-Walker and Burg methods for fitting AR($p$) models to data, by generating data from a specific AR(2) model and fitting models of that order with these two methods. No clear differences in the ability to make one-step predictions were noted; neither was the hypothesis that the residuals from these models should have the same distribution as those of the innovations in the true data generating process. We also compared 1- and 3-step predictions from an AR(10) model fitted to the same data; this one was much worse in comparison based on the higher mean square prediction error.

# Problem 4

## Objectives

We now consider the causal AR(1) process

$$X_t = 0.8X_{t-1} + Z_t, \quad \{Z_t\} \sim \text{WN}(0, 1) \tag{2}$$

Such a process may also be represented by an infinite order moving average process. We are to simulate 100 samples of size 200 from this model, with iid normal $Z_t$s. Our objectives are to fit both an AR(1) model and a MA(10) model to the data and compare how well the two fitted models do one-step and three-step predictions.

## Methodology and mathematical background

In problem 3 we covered two different methods for fitting AR(p) models to data; here we will use the Yule-Walker equations for fitting the AR(1) model. To fit an MA(10) model, we will use the **innovations algorithm**, as covered in section 2.5.2 of [1, pp. 71–75]. Having no desire to rewrite the matrix expressions of that section, I skip to the punchline: for a zero-mean series with finite second moment and covariance $\text{E}[X_i X_j] = \kappa(i, j)$, the moving average coefficients $\theta_{n1}, \ldots, \theta_{nn}$ can be computed recursively from

$$\begin{aligned}
\upsilon_0 &= \kappa(1, 1) \\
\theta_{n,n-k} &= \upsilon_k^{-1} \left( \kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \theta_{n,n-j} \upsilon_j \right), \quad 0 \leq k \leq n \\
\upsilon_n &= \kappa(n+1, n+1) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 \upsilon_j
\end{aligned}$$

where $\upsilon_n = \text{E}[(X_{n+1} - P_n X_{n+1})^2]$ is the expected squared one-step prediction error, which approaches the white noise variance as $n$ increases. In a practical setting, the components in the above algorithm are replaced by their sample equivalents [1, pp. 150–151]. However, while $\hat{v}_n$ is consistent, the MA coefficient sample estimates are not [1, p. 151].

## Results

In figure 12 we show a histogram of the one-step prediction errors of the fitted AR(1) and MA(10) models.

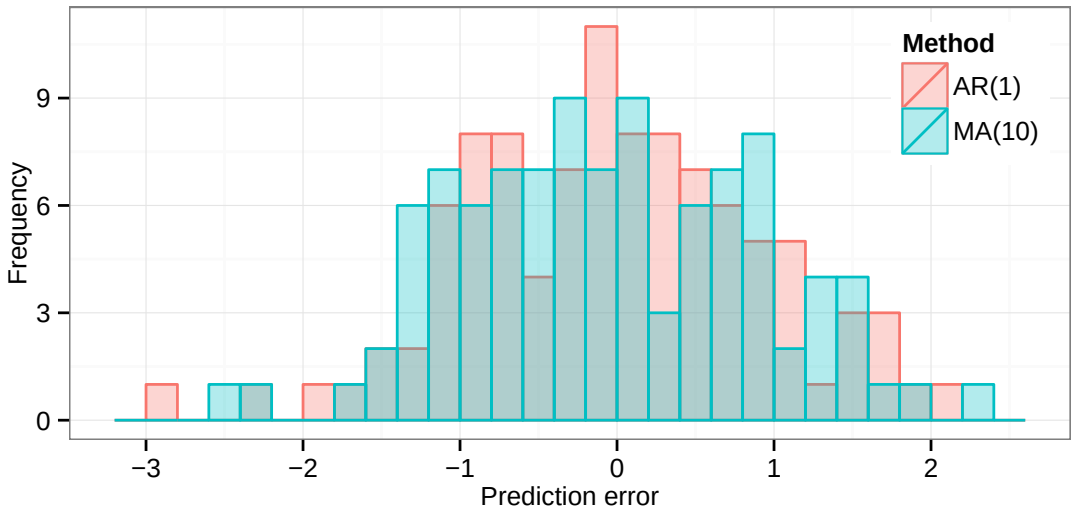


Figure 12: One-step prediction errors for the fitted AR(1) and MA(10) models.

Both models seem to produce prediction errors fairly symmetric about 0. The mean one-step prediction error of the fitted AR(1) models is -0.03 and the median is -0.04; the corresponding numbers for the fitted MA(10) models are -0.06 and -0.1. The one-step mean square prediction error for the AR(1) model is estimated to be 0.87, and for the MA(10) model the estimate is 0.9. The AR(1) model—the same model as for the true data generating process—thus seems to be preferable for prediction.

Figure 13 shows the 3-step predictions for the two fitted models. Again the histograms are fairly similar, though the errors are evidently larger in magnitude compared to the one-step prediction errors.

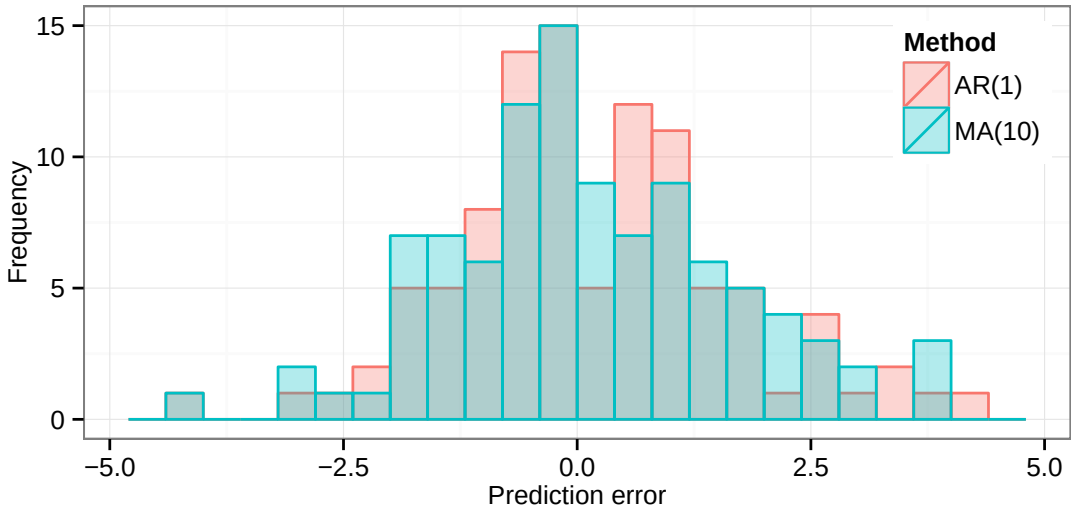


Figure 13: Three-step prediction errors for the fitted AR(1) and MA(10) models.

The mean three-step prediction error of the fitted AR(1) models is 0.13 and the median is -0.08; the corresponding numbers for the fitted MA(10) models are 0.12 and -0.1. The one-step mean square prediction error for the AR(1) model is estimated to be 2.21, and for the MA(10) model the estimate is 2.37. Again the AR(1) model seems preferable for prediction purposes.

## Summary

Knowing that a causal AR(1) process can be represented as an infinite order MA process, we have generated data from the AR(1) model (2), and fitted an AR(1) and a MA(10) model to this data (over several samples). We then compared how well these models do at prediction, by comparing their prediction errors and mean square prediction errors. The results did not differ much, though the fitted AR(1) models seemed slightly better—and is furthermore preferable due to its simplicity, in comparison to the MA(10) model.

# Problem 5

## Objectives

We are again considering model (2), this time with the noise variables $\{Z_t\} \sim \text{WN}(0, 1)$ such that $c + Z_t$ are lognormally distributed for an appropriate choice of $c$ and parameters of the lognormal distribution. After figuring out what these appropriate choices are, we are to simulate 100 samples of size 200 from this model, and for each sample fit an AR(1) model to the data. We will make a scatterplot of the estimated parameter pair consisting of the AR(1) coefficient and the standard deviation of the white noise sequence. Lastly, we will make one-step predictions and compare the results to those found in Problem 4.

## Methodology and mathematical background

Let $\log(c + Z_t)$ be a normally distributed random variable with mean $\mu$ and variance $\sigma^2$. The constants $c$, $\mu$ and $\sigma^2$ are to be chosen so that $Z_t$ has the properties of the noise in model (2), i.e. has expected value zero and variance 1. $c + Z_t$ is a lognormally distributed random variable with mean $e^{\mu+\sigma^2/2}$ and variance $(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$. Choosing $\sigma^2 = \log 2$, its variance becomes $2e^{2\mu}$. Then taking $\mu = -\frac{1}{2}\log 2$, its variance is 1, and its mean becomes $e^{-\frac{1}{2}\log 2+\frac{1}{2}\log 2} = e^0 = 1$. Choosing $c = 1$, we see that $\text{E}[Z_t] = \text{E}[Z_t + c] - c = 1 - 1$, and $Z_t$ thus has the required properties.

To simulate from this distribution, we can use the following function:

```
# Simulate from shifted lognormal distribution with mean 0 and variance 1
zdist <- function(n) {
  z <- rlnorm(n, meanlog = -log(2) / 2, sdlog = sqrt(log(2))) - 1
  return(z)
}
```

which uses R's built-in random number generator `stats::rlnorm` for the log-normal distribution. To fit the AR(1) model (2), we again use the Yule-Walker equations as described in Problem 3, also using the same prediction methodology and way of estimating the MSPE.

## Results

In figure 14 we show the estimated noise standard deviation $\sigma$ plotted against the estimated AR(1) coefficient $\phi$. In the true model, these are 1 and 0.8 respectively.

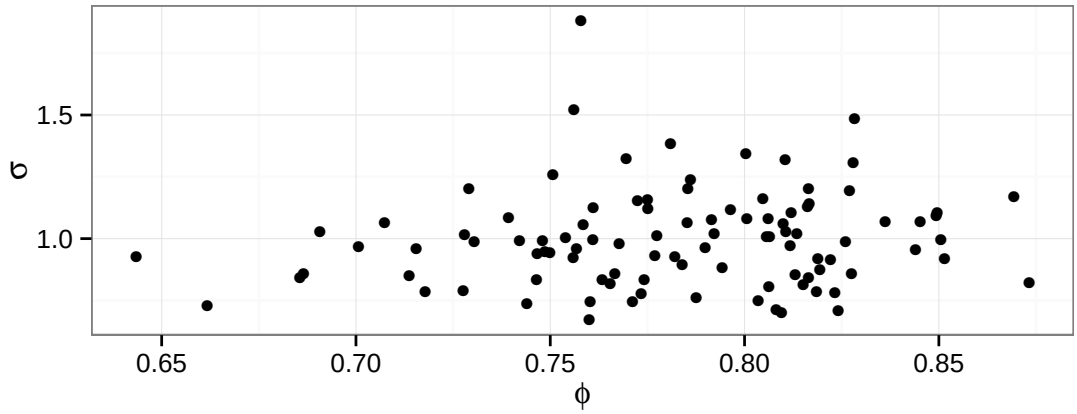


Figure 14: Estimated noise standard deviation vs AR(1) coefficient for model (2), fitted by the Yule-Walker equations.

There is no obvious relation between the estimated parameter pairs, though there may be a slight negative skew in the sample distribution of the estimated AR(1) coefficient, and a positive skew in that of the estimated noise standard deviation. Figure 15 illustrates this more clearly:

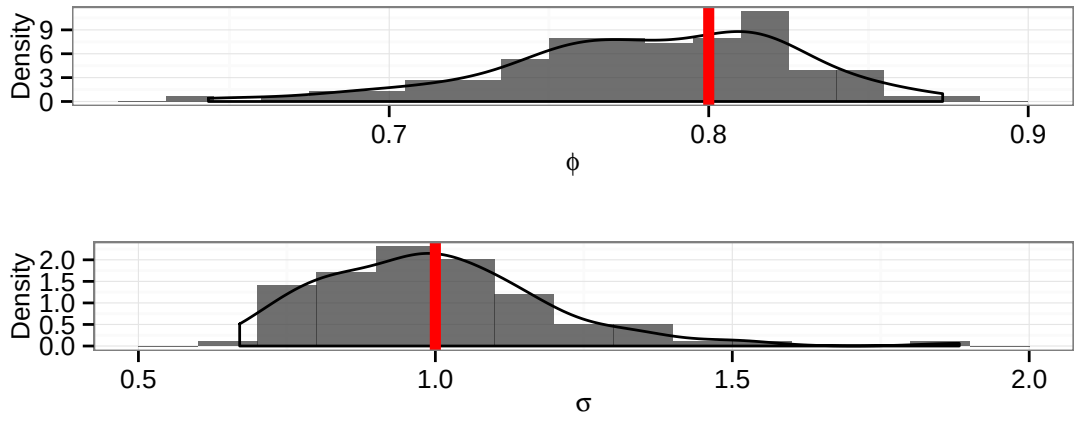


Figure 15: Density plot of estimated AR(1) coefficient and noise standard deviation for model (2), fitted by the Yule-Walker equations. Red vertical line marks the true parameter value.

Moving on to the one-step predictions, figure 16 shows the prediction errors of the fitted models.

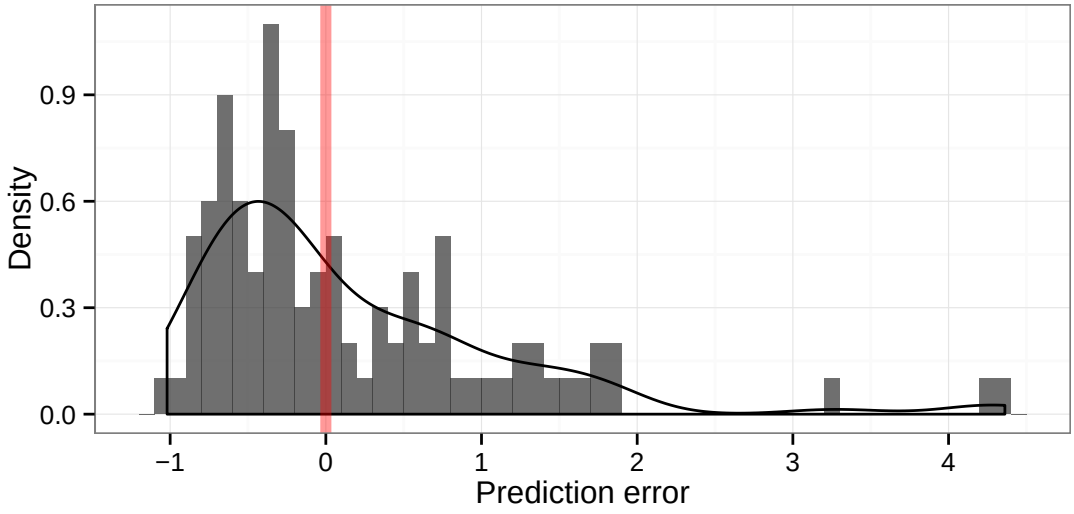


Figure 16: Prediction errors for model (2) with (shifted) lognormal noise, fitted by the Yule-Walker equations. The red line indicates the best possible error, equal to zero.

As can be seen, the prediction errors are positively skewed, with a mean of $-0.15$ and a median of $-0.23$. The one-step MSPE is 1.01, which is seemingly larger than the corresponding MSPEs for the MA(10) and AR(1) models fitted in Problem 4. However, also recall that in that problem, we used another estimation method (Burg's algorithm). Further, the true MSPE is the variance of the innovations, 1 in this case, and here the model in question 5 comes closer. Also, we saw in Problem 4 that the bulk of prediction errors are between are less than two in magnitude, found in the interval between $-2$ and 2, while here in Problem 5 most prediction errors are less than 1 in magnitude, found in the smaller interval $-1$ to 0. So in Problem 5, the predictions often overestimates the next value of the process, though underestimations are often larger in magnitude. Naturally, this has to do with the distribution of the noise sequence being skewed the way it is.

### Summary

We have fitted an AR(1) model to data generated from such a process, with the noise terms having a shifted lognormal distribution with mean zero and unit variance. We found that prediction errors were smaller compared to those in Problem 4, though different estimation methods were used there, and the noise distribution was also different.

# Problem 6

## Objectives

In this problem we are given six time series of unknown origin. We are to find and remove any trend or seasonal components from these series, and then choose a stationary time-series model for the transformed data. Choosing either AR($p$), MA($q$) or ARMA($p, q$) models and their orders, we are to estimate the parameters of these models using two different approaches. We will then compare the results, and discuss why one of them would be preferable to the other.

## Methodology and Mathematical Background

Following section 1.5 of [1], our first action for a given data set will be to plot the data and look for signs of sudden shifts in behavior of the series, suspicious outliers, or if we can otherwise spot patterns that resemble known deterministic functions. The latter might suggest that the process could be represented by the **classical decomposition model**

$$X_t = m_t + s_t + Y_t \tag{0.21}$$

where the trend component $m_t$ is a slowly changing deterministic function, the seasonal component $s_t$ is a deterministic function such that $s_{t+d} = s_t$ for a period $d$ (which may or may not be known), and $Y_t$ is a stationary random noise component with zero mean. Possibly, we may have to transform the data in some way (e.g. by taking logarithms) before obtaining this representation. Another approach, which may be necessary if we only want to fit stationary time series models, is to first difference the data sufficiently many times before fitting such models. We will take a model-builidng approach below, focusing on parametric representations for the trend and seasonal components. For example, trend components may be estimated by fitting polynomials, and seasonal components by fitting sums of trigonometric functions (sines and cosines) with given periods. Either approach allows us to forecast

Following sections 1.5, 1.6, 5.3, 5.5 and 6.2 of [1], our general modeling methodology will be as follows:

1. Plot the raw data and see if there are any clear trend or seasonal components, or if some type of transformation (e.g. logarithm) or differencing needs to be made.
2. If a model seems appropriate for a trend (typically a polynomial), fit it to the data and inspect the residuals. If we have not made things worse, proceed.
3. Inspect the sample spectral density and ACF of the detrended data to get a sense of any seasonality (and its period).
    - If there is seasonality, a good first choice will be to remove it by fitting a sum of sine and cosine terms with the indicated period to the detrended data. Then inspect the spectral density and ACF again to see if it made a difference.
    - Knowledge of the type of data (e.g. monthly data from an environmental process) can help greatly in finding the correct period. Unfortunately, we know nothing of the data in this problem.
4. If satisfied that periodicity is removed, reestimate trend and seasonal components together.

5. If the residuals after removing trend and seasonality seem stationary (which we judge by plotting them and looking at the sample ACF and partial ACF), we
    - Look at the sample ACF and PACF to see if there are sharp cut-offs in one of them and (perhaps alternating) exponential decay in the other. I.e. we look for characteristic patterns of pure AR($p$) or pure MA($q$) processes.
    - However, as noted in [1, p. 155], if both $p$ and $q$ are greater than zero, then the sample ACF and PACF are not as indicative of the model order. In that case, we fit ARMA models of different orders and choose the one which minimizes the AICC. In that case:
        - Order selection by minimization of the AICC is done through fitting models by maximum likelihood, as stated in [1, p. 155]. However, as I interpret that statement and more so how the instructions for Problem 6 reads (that we need to use two different estimation methods), after deciding model order the method of parameter estimation need not be limited to maximum likelihood. In any case, the differences are not likely to be large.
        - As stated on page 180 of [1], one should not blindly choose the model with the smallest AICC if the differences in it between the top contending models is small. In this case, choosing the simplest model would be preferable.
6. After choosing the appropriate model for the detrended and deseasonalized data, we fit the model with the method(s) of our choosing and inspect the residuals. Our hope is that these residuals resemble a white noise or iid sequence, and so we use the tests and methods discussed in Problem 1 and in section 1.6 of [1] to see if this is so.
7. At any step (particularly the previous one) we may see reason to return to a previous step of the modeling process and make a different choice. We must also realize that the tools we are equipped with may not allow us to solve every problem to full satisfaction (as a hint to problems 7 and 8).

We briefly also introduce two more algorithms or estimation methodologies, intended for the fitting of ARMA$(p, q)$ models $\phi_p(B)X_t = \theta_q(B)Z_t$. The first is the **Hannan-Rissanen algorithm**, which uses least-squares linear regression by [1, pp. 156–157]

1. Fitting a high order AR($m$), $m > \max(p, q)$ using the Yule-Walker equations, and extracting the residuals $\hat{Z}_t, t = m + 1, \ldots, n$.
2. Estimating the parameter vector $\boldsymbol{\beta} = (\boldsymbol{\phi}', \boldsymbol{\theta}')'$ by linear regression of $X_t$ onto $(X_t - 1, \ldots, X_{t-p}, Z_{t-1}, \ldots, Z_{t-q})$, choosing the parameters which minimizes the sum of squared errors.
3. The estimate of the white noise variance is that sum divided by $n - m - q$.

The second method is **maximum likelihood estimation**, using a Gaussian likelihood for $\{X_t\}$ even if that may not be the case in the true process, as the MLEs of the ARMA coefficients still have the usual asymptotic normal distribution [1, p. 159].

### Results: Dataset 1

In figure 17 we see the first dataset given. Apart from a possible slight upward trend, it is hard to notice any clear seasonal components.

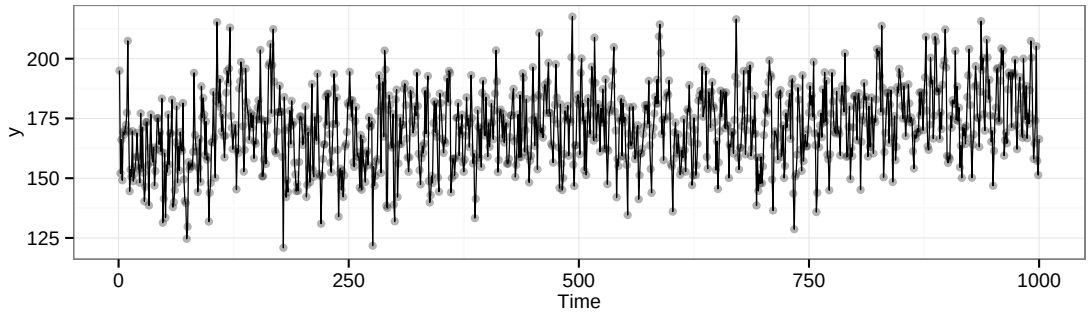


Figure 17: Dataset 1 of Problem 6.

We fit a linear trend component to the data with `stats::lm`, yielding the slope and intercept

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 163.65 | 0.98 | 166.37 | 0.00 |
| t | 0.02 | 0.00 | 8.88 | 0.00 |

with an adjusted $R^2$ value of 0.07. To determine if there are any seasonal components, we plot the spectral density in figure 18 on the left. As can be seen, there are two distinctive peaks, the taller one indicating a period of 12. We then fit (with `stats::lm`) a model consisting of the sum of one sine and one cosine component to the detrended data, each with period 12. It turns out the coefficient (amplitude) of the cosine component is not significantly different from zero, so we refit the model leaving the cosine (and intercept) part out. Plotting the spectral density again, shown on the right of figure 18, we see that a period of 6 remains. Removing this period in the same manner, the sample spectral density will no longer have a peak away from zero, giving us some indication of successful removal of seasonality.

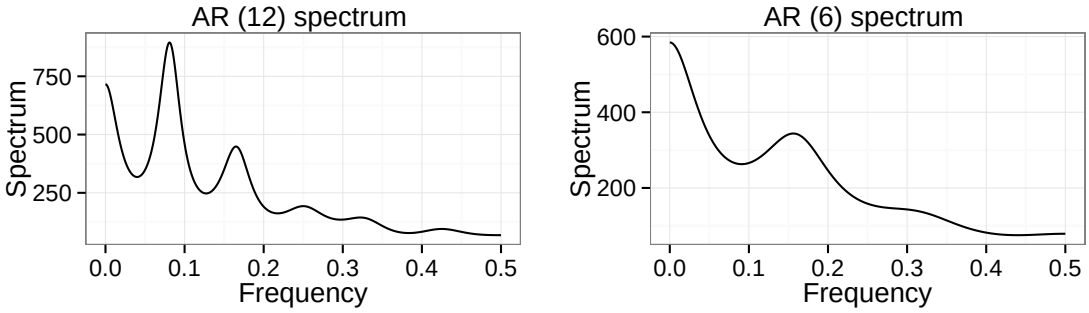


Figure 18: Spectral density of detrended data (left) and detrended data with a harmonic component (sine) of period 12 removed.

We then refit the trend and seasonal components together, giving us the model

```
Call:
lm(formula = y ~ 1 + t + sin(2 * pi * t/12) + sin(2 * pi * t/6) +
    cos(2 * pi * t/6), data = d1)

Residuals:
   Min     1Q Median     3Q    Max
-50.21  -8.83  -0.40   8.18  48.05

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        163.68221    0.89258  183.38  < 2e-16 ***
t                    0.01512    0.00154    9.79  < 2e-16 ***
sin(2 * pi * t/12)  -8.14617    0.63021  -12.93  < 2e-16 ***
sin(2 * pi * t/6)    1.68592    0.63052    2.67   0.0076 **
cos(2 * pi * t/6)    4.10539    0.63083    6.51  1.2e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.1 on 995 degrees of freedom
Multiple R-squared:  0.239,Adjusted R-squared:  0.236
F-statistic: 78.2 on 4 and 995 DF,  p-value: <2e-16
```

We now hope that the residuals after removing trend and seasonal components from the original data are stationary. We plot them in figure 19 below; nothing seems out of the ordinary (stationary) except perhaps that there is a bit more spread in the early parts of the series.

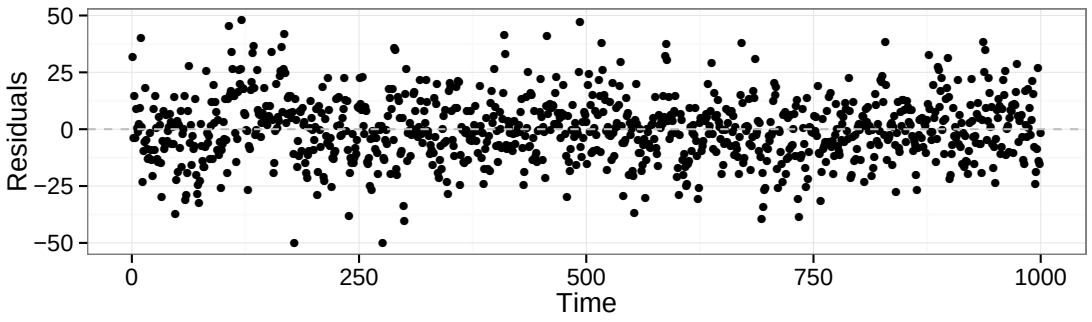


Figure 19: Residuals after removing trend and seasonal components from raw data.

In figure 20, we plot the sample ACF and PACF. With a spike at lag 1 in the PACF and a decaying ACF, these plots indicate, in addition to stationarity of the residuals, that a possible model for the noise sequence (residuals) is an autoregressive model of order 1 (with zero mean, due to dealing with residuals).

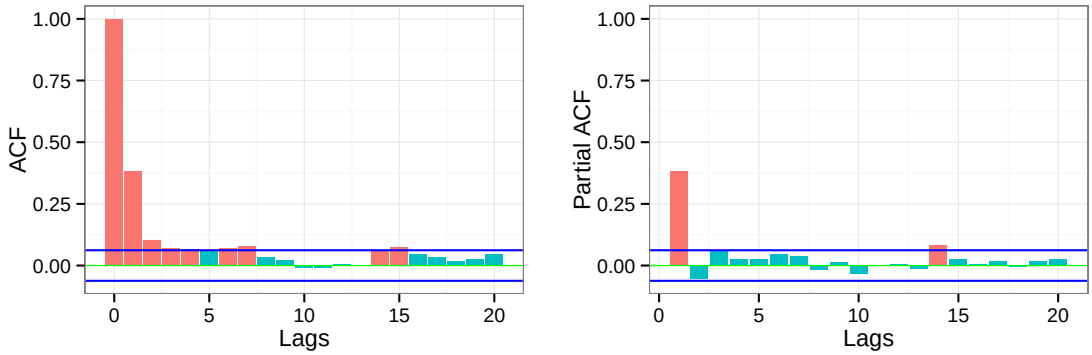


Figure 20: Sample ACF and PACF of residuals from removing trend and seasonal components from the raw data.

Finding the simplicity of the AR(1) model alluring, we fit it to the noise to see if the residuals from that model, in turn, are likely to be white noise or even iid. For this purpose we use the Yule-Walker equations and Burg's algorithm, as covered earlier. The particular R functions are as mentioned `stats::ar.yw` and `stats::ar.burg`. The estimated AR coefficient of the Yule-Walker model is 0.38 and the estimated variance of the noise is 169.17. For this fitted model, we run the tests covered in Problem 1, now using the function `itsmr:test` to get the aggregated results:

```
Null hypothesis: Residuals are iid noise.
Test                         Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)      21.4    0.3737
McLeod-Li Q                 Q ~ chisq(20)     93.56         0 *
Turning points T (T-664.7)/13.3 ~ N(0,1)       670    0.6887
Diff signs S        (S-499)/9.1 ~ N(0,1)       514    0.1003
Rank P      (P-249250.5)/5266.5 ~ N(0,1)    250203    0.8565
```

As can be seen, the null hypothesis of iid noise fails to be rejected for every test except for the McLeod-Li test. Nevertheless, the results are indicative that we have found a good model.

To be on the safe side, we also make a few diagnostics plots in figure 21. There is nothing too alarming in these plots, although the spikes at lag 2 in both the ACF and PACF are perhaps a bit worrisome.

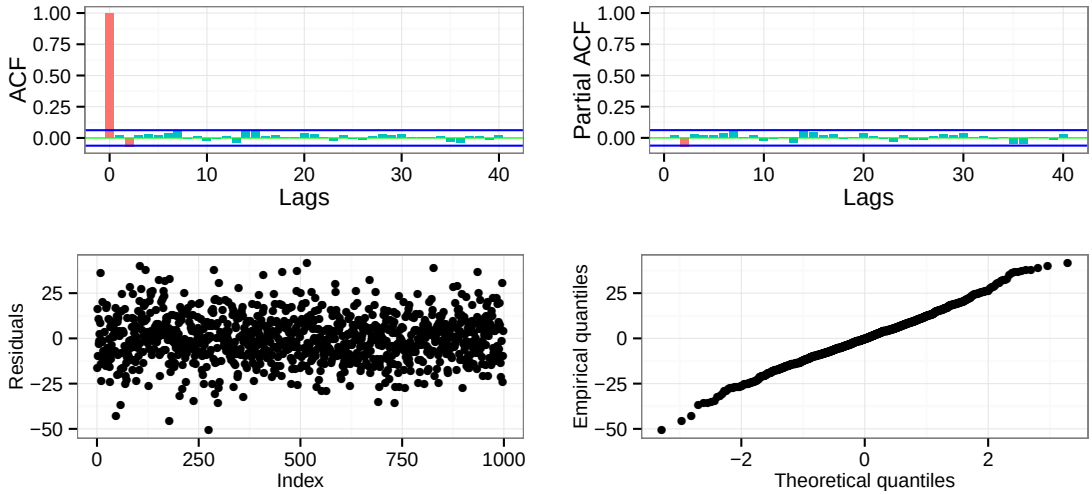


Figure 21: Diagnostics plots for the AR(1) model fitted by the Yule-Walker equations.

For Burg's method, estimate of the AR(1) coefficient is 0.38 and the estimate of the noise variance is 168.68. As can be seen, the difference compared to the Yule-Walker results is negligible. Running the same tests as for the Yule-Walker model, we see that the results also match:

```
Null hypothesis: Residuals are iid noise.
Test                         Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     21.39    0.3747
McLeod-Li Q                 Q ~ chisq(20)     93.49         0 *
Turning points T (T-664.7)/13.3 ~ N(0,1)        670    0.6887
Diff signs S        (S-499)/9.1 ~ N(0,1)        514    0.1003
Rank P      (P-249250.5)/5266.5 ~ N(0,1)     250205    0.8562
```

The diagnostics plots for Burg's method are almost identical to those above, so we omit them here.

For this dataset, an AR(1) model seemed appropriate for the data after removing trend and seasonal components. We used the Yule-Walker equations and Burg's algorithm to fit this model, and the results were qualitatively identical. As such, we see no indication that one method would be preferable to the other. Had we been tasked with prediction, our results and conclusions may have been different. The model for the data is, finally

$$
\begin{aligned}
Y_t &= m_t + s_t + X_t, \quad \text{with} \\
\hat{m}_t &= 163.7 + 0.02t \\
\hat{s}_t &= -8.1\sin(2\pi t/12) + 1.7\sin(2\pi t/6) + 4.1\cos(2\pi t/6) \\
X_t &= 0.38X_{t-1} + Z_t, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{WN}(0, 169)
\end{aligned}
$$

### Results: Dataset 2

The next dataset is given as in figure 22. Similar to the previous dataset, there is no obvious trend, seasonal component or other type of dependency in the data, except perhaps a slight upward linear trend.

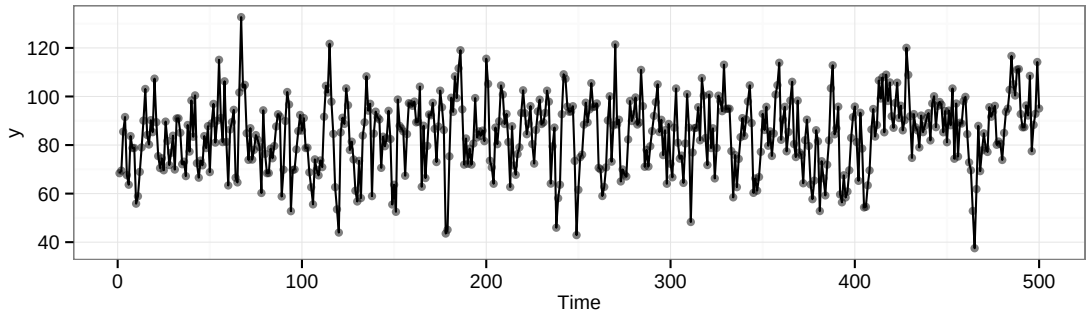


Figure 22: Dataset 2 of Problem 6.

We fit and remove such a trend from the data, obtaining the estimates

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 80.77 | 1.32 | 61.23 | 0.00 |
| t | 0.01 | 0.00 | 3.03 | 0.00 |

which are seen to be significantly different from zero, but with an adjusted $R^2$ value of only about 0.01. Turning to possible seasonal components, we plot the sample spectral density of the detrended data in figure 23, on the left.

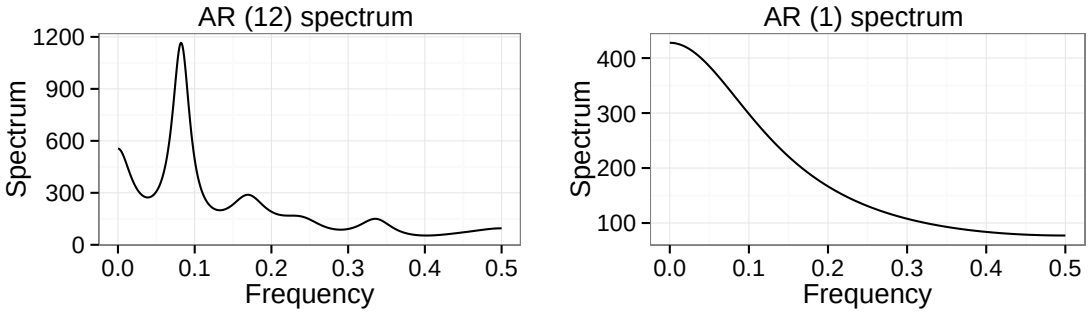


Figure 23: Spectral density of detrended data (left) and detrended data with a harmonic component (sum of sine and cosine) of period 12 removed.

As can be seen, the peak of the function corresponds to a period of 12, and we therefore fit to the detrended data a sum of a sine and cosine term with that period. In figure 23, on the right, we plot the spectral density of the residuals from that model. As can be seen, the periodicity seems to have disappeared.

We now refit the trend and seasonal components together, giving the following model:

```
Call:
lm(formula = y ~ 1 + t + sin(2 * pi * t/12) + cos(2 * pi * t/12),
    data = d2)

Residuals:
   Min     1Q Median     3Q    Max
-44.92  -8.30   0.61   8.35  47.42

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        80.73752    1.21032   66.71  < 2e-16 ***
t                   0.01377    0.00419    3.29   0.0011 **
sin(2 * pi * t/12)  4.72229    0.85455    5.53  5.3e-08 ***
cos(2 * pi * t/12) -6.90994    0.85455   -8.09  4.8e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.5 on 496 degrees of freedom
Multiple R-squared:  0.177,Adjusted R-squared:  0.172
F-statistic: 35.5 on 3 and 496 DF,  p-value: <2e-16
```

As seen, the adjusted $R^2$ is much improved from just the linear model, and all coefficients are significantly different from zero at standard significance levels. We will now inspect the residuals from this model to see if they resemble a stationary sequence, in which case we will try to find the appropriate model for them.

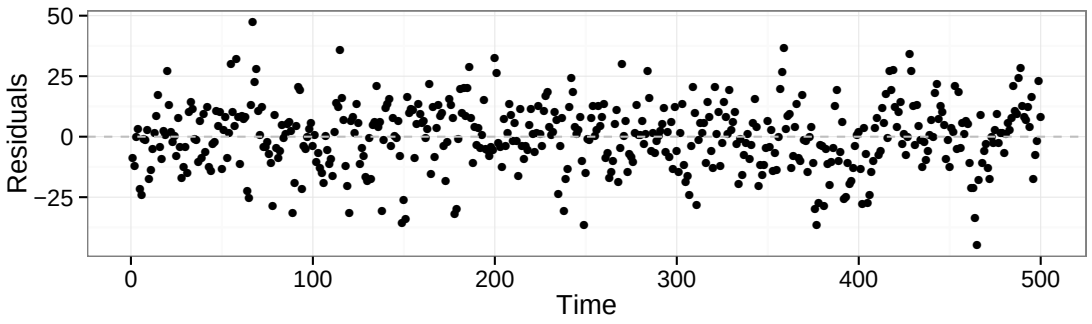


Figure 24: Residuals after removing trend and seasonal components from raw data.

Figure 24 gives no particular indication to doubt that the residuals are stationary, and we move on to plot the sample ACF and PACF.

Similar to the first dataset of Problem 6, the ACF and PACF of the detrended and deseasonalized series shown in figure 25 have a spike at lag 1 in the PACF and an ACF that decays quickly, suggesting that an AR(1) model may be appropriate.

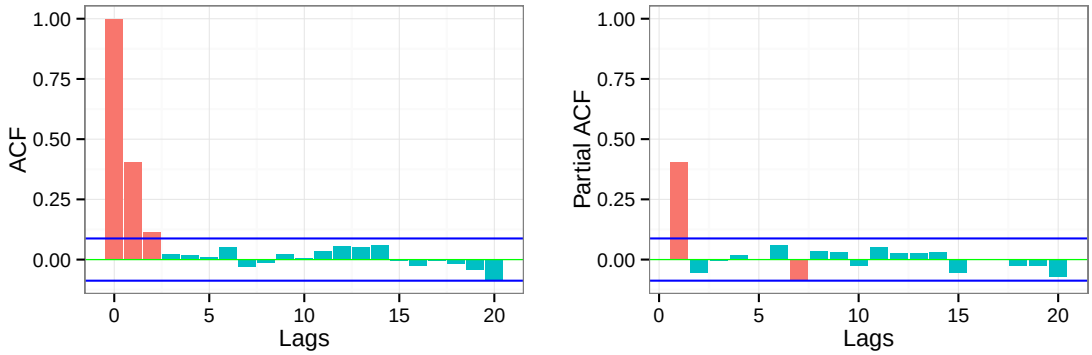


Figure 25: Sample ACF and PACF of residuals from removing trend and seasonal components from the raw data.

With no indication that the (zero-mean) AR(1) model would be a particularly bad choice, we continue by fitting such a model to the residuals, again using the Yule-Walker equations and Burg's algorithm. The estimated AR coefficient of the Yule-Walker model is 0.4 and the estimated variance of the noise is 152.21. Also running the same tests as before yields

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)     13.05    0.8752
McLeod-Li Q                Q ~ chisq(20)     46.59     7e-04 *
Turning points T  (T-331.3)/9.4 ~ N(0,1)       327    0.6449
Diff signs S        (S-249)/6.5 ~ N(0,1)       262     0.044 *
Rank P       (P-62125.5)/1860.6 ~ N(0,1)     62184    0.9749
```

Again the McLeod-Li test shows a low $p$-value, but on the other hand the Ljung-Box test $p$-value is higher in comparison to those found for the previous dataset. The difference-sign test also rejects the null hypothesis of iid noise, though as we noted in Problem 1 the performance of this test was not always stellar.

We can get a further indication of how good the AR(1) model is by looking at diagnostics plots of the residuals from the fitted model, as shown in figure 26

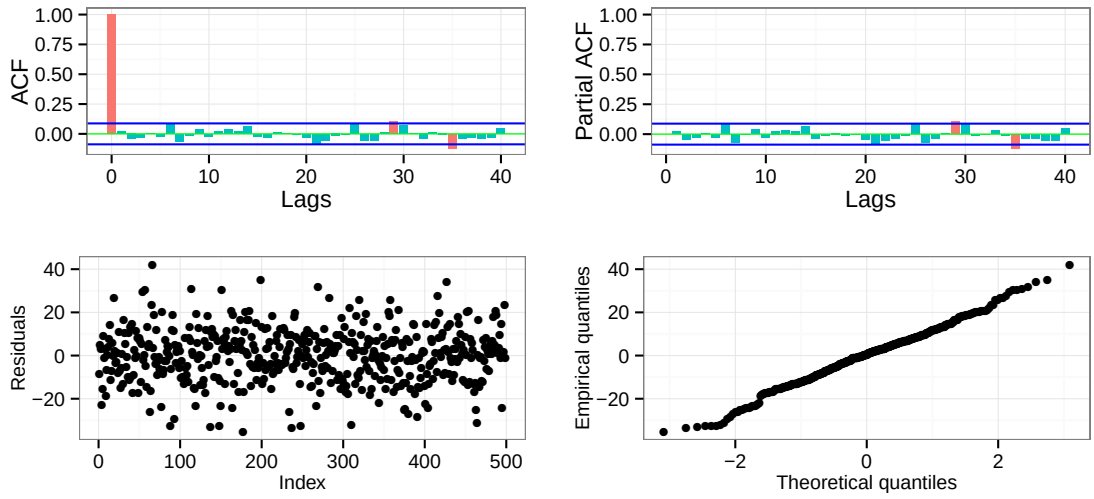


Figure 26: Diagnostics plots for the AR(1) model fitted by the Yule-Walker equations.

Nothing in these plots is particularly alarming, except perhaps the sudden leftward shift in the left tail of the empirical distribution of the residuals.

For Burg's method, estimate of the AR(1) coefficient is 0.4 and the estimate of the noise variance is 151.55. As can be seen, the difference compared to the Yule-Walker results is negligible. Running the same tests as for the Yule-Walker model, we see that the results also match:

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)     13.05     0.875
McLeod-Li Q                Q ~ chisq(20)     46.61     7e-04 *
Turning points T  (T-331.3)/9.4 ~ N(0,1)       327    0.6449
Diff signs S        (S-249)/6.5 ~ N(0,1)       262     0.044 *
Rank P       (P-62125.5)/1860.6 ~ N(0,1)     62174    0.9792
```

The diagnostics plots are again almost identical, and are omitted. What can be noted in comparison to the previous dataset where we also fitted an AR(1) model, is that again the Yule-Walker equations delivered a higher estimated noise variance.

For this dataset, an AR(1) model seemed appropriate for the data after removing trend and seasonal components. We used the Yule-Walker equations and Burg's algorithm to fit this model, and the results were qualitatively identical. As such, we see no indication that one method would be preferable to the other. Had we been tasked with prediction, our results and conclusions may have been different. The model for the data is, finally

$$\begin{aligned} Y_t &= m_t + s_t + X_t, \quad \text{with} \\ \hat{m}_t &= 80.7 + 0.01t \\ \hat{s}_t &= 4.7\sin(2\pi t/12) - 6.9\cos(2\pi t/12) \\ X_t &= 0.4X_{t-1} + Z_t, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{WN}(0, 152) \end{aligned}$$

### Results: Dataset 3

The next dataset is shown in figure 27. As is obvious, there is a clear cyclical component in the data, with a period that is somewhere in the range 100–200.

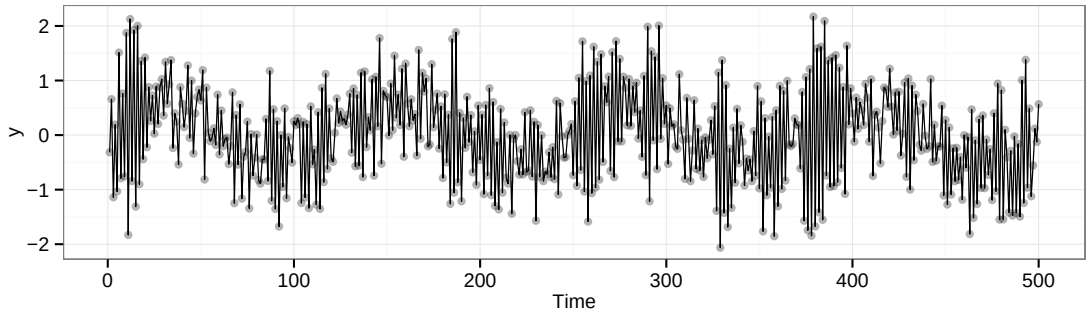


Figure 27: Dataset 1 of Problem 6.

No trend is apparent in the data, and fitting a linear model to it yields a slope coefficient that is negligible in size. Instead, we focus on removing the obvious seasonal component, by first finding the period of this seasonality. Unfortunately, the sample spectral density or ACF is not of much use here, because the period is too large in relation to the sample size. Instead, we use an ad-hoc approach of fitting a model of the form $y = a\sin(2\pi t/k) + b\cos(2\pi t/k)$ to the data, for $k = 100, 101, \ldots, 200$, and choose the period $k$ that yields the largest adjusted $R^2$ value, using the following procedure:

```
periods.d3 <- 100:150
r2s.d3 <- rep(NA, length(periods.d3))
for (i in seq_along(periods.d3)) {
#   r2s.d3[i] <- summary(lm(y ~ 0 + sin(2*pi*t / periods.d3[i])
#               + cos(2*pi*t / periods.d3[i]), data=d3))$adj.r.squared
  r2s.d3[i] <- summary(lm(y ~ 0 + sin(2*pi*t / periods.d3[i]),
                          data=d3))$adj.r.squared
}
# Best period equals 126 with commented out model, but cosine term
# is not significantly different from zero. Removing this component
# and re-running the procedure yields the best period 125
period.d3 <- periods.d3[which.max(r2s.d3)] #  == 125
```

This yields a model without a cosine term, and period of 125, shown on the next page

```
Call:
lm(formula = y ~ 0 + sin(2 * pi * t/125), data = d3)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0944 -0.5332  0.0087  0.5959  2.0713

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
sin(2 * pi * t/125)   0.5016     0.0496    10.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.784 on 499 degrees of freedom
Multiple R-squared:  0.17,Adjusted R-squared:  0.169
F-statistic:  102 on 1 and 499 DF,  p-value: <2e-16
```

We also investigated the residuals from this model, shown in figure 28, for any types of trend components, but found none.

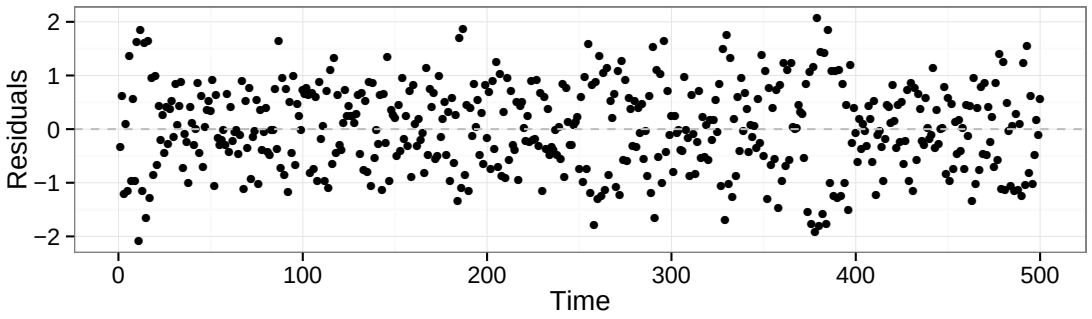


Figure 28: Residuals after removing a long seasonal component from the raw data.

We therefore look to see if these residuals form a stationary sequence which we can model with an AR, MA or ARMA model. The plot above indicates that they are stationary, in the sense that there is no clear dependency and run-off variance. In figure 29 we plot the sample ACF and PACF of the residuals of the above model.

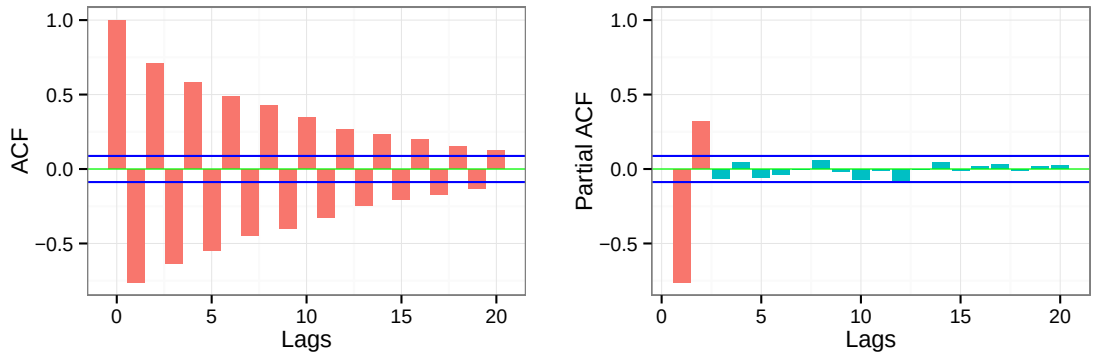


Figure 29: Sample ACF and PACF of residuals from removing long seasonal component from the raw data.

Though the PACF does not look dissimilar to those we (I) may have seen from AR models on other occasions, the sample ACF clearly has a peculiar behavior, though at least it seems to decay at an (alternating) exponential pace. Therefore we will not base order selection on the looks of these plots, but will rather choose the ARMA model that minimizes the AICC. We restrict the search to zero-mean models, based on the residual plot 28. The R function `forecast::auto.arima` automates this procedure for us, fitting models with maximum likelihood and selecting the one which minimizes the AICC. This turns out to be an ARMA(1,1) model, which seems a parsimonious choice. The model fitted by maximum likelihood is

```
Series: d3$stationary
ARIMA(1,0,1) with zero mean

Coefficients:
         ar1    ma1
      -0.919  0.416
s.e.   0.021  0.049

sigma^2 estimated as 0.232:  log likelihood=-344.8
AIC=695.6   AICc=695.7   BIC=708.3
```

Again we run the tests discussed in Problem 1 on the residuals of this model, to ensure that the model decision is appropriate.

The results of these tests for the ML-fitted model are

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     12.89    0.8819
McLeod-Li Q                 Q ~ chisq(20)     11.37    0.9361
Turning points T    (T-332)/9.4 ~ N(0,1)       342     0.288
Diff signs S      (S-249.5)/6.5 ~ N(0,1)       251    0.8164
Rank P         (P-62375)/1866.2 ~ N(0,1)     58874    0.0607
```

As can be seen, the tests give no indication that the residuals are not iid, though the $p$-value for the rank test is fairly small. Neither do the diagnostics plots, seen in figure 30, give any indication that the model is particularly wrong. We therefore consider it a good choice.

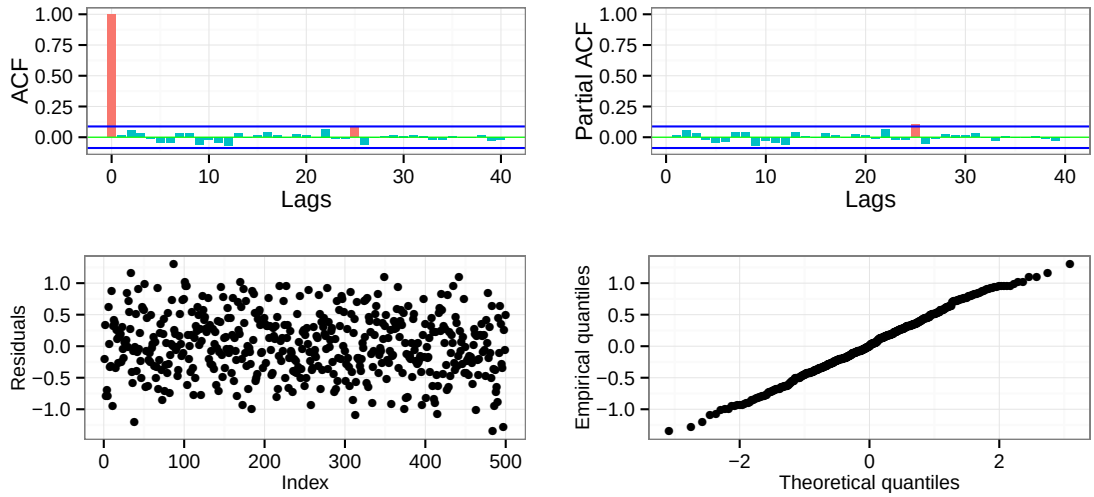


Figure 30: Diagnostics plots for the ARMA(1,1) model fitted by maximum likelihood.

We also fit the ARMA(1,1) model using the Hannan-Rissanen algorithm, which yields estimates

```
    phi    theta   sigma2   se.phi se.theta
  -0.90     0.42     0.23     0.04     0.06
```

The test results and diagnostics plots are very similar to those of the maximum likelihood model, showing no qualitative differences. Based on this, it is hard to say that one approach is better than the other. However, the maximum likelihood estimates did give smaller standard errors for the the model coefficients, perhaps tilting the preference in its favor. The model for the data is, finally

$$
\begin{aligned}
Y_t &= s_t + X_t, \quad \text{with} \\
\hat{s}_t &= 0.5 \sin(2\pi t/125)) \\
X_t &= -0.9X_{t-1} + Z_t + 0.4Z_{t-1}, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{WN}(0, 0.23)
\end{aligned}
$$

### Results: Dataset 4

As can be seen from figure 31, the data for this problem seems to have an upward trend from about the 150th observation and forward, suggesting that a simple linear model is insufficient.

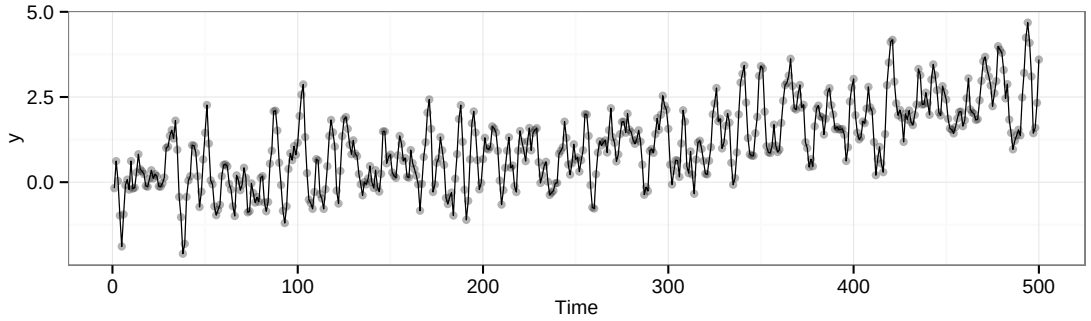


Figure 31: Dataset 4 of Problem 6.

Instead, we try to find a trend component in the form of a second degree polynomial. It turns out that the best-fitting such polynomial is one without intercept or linear term, given as follows:

```
Call:
lm(formula = y ~ 0 + I(t^2), data = d4)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1175 -0.4999  0.0494  0.6185  2.7427

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
I(t^2) 1.20e-05   3.36e-07    35.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.842 on 499 degrees of freedom
Multiple R-squared:  0.718,Adjusted R-squared:  0.717
F-statistic: 1.27e+03 on 1 and 499 DF,  p-value: <2e-16
```

As seen, the fitted trend has quite high adjusted $R^2$, suggesting that it can explain much of the variability in the dependent variable.

In figure 32, we show the residuals of the fitted trend component consisting of a single quadtratic component.

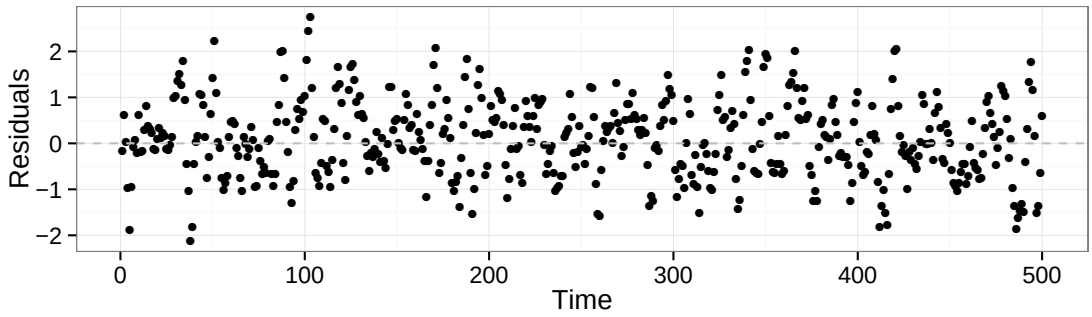


Figure 32: Residuals after removing trend component from raw data.

There are no obvious signs of seasonality in the above residuals, but we investigate it nonetheless by plotting the sample spectral density in figure 33.

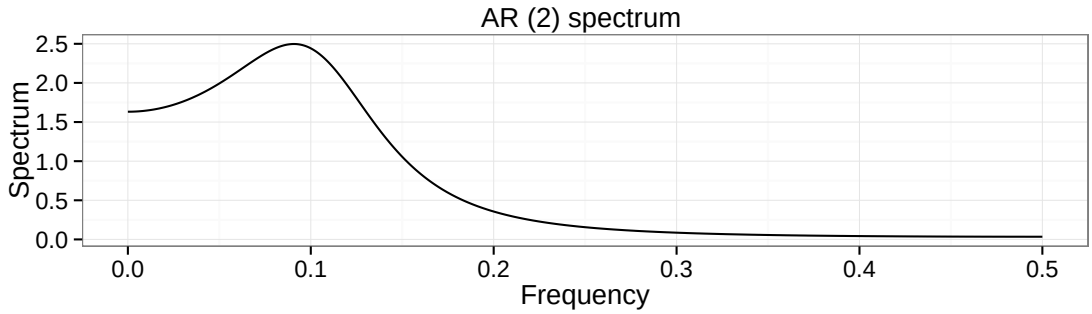


Figure 33: Spectral density of detrended data.

If interpreted as before, the spectral density would suggest a period of length 2. However, it seems more likely that this is indicative of an autoregressive component of order two. As the residuals in figure 32 show no signs of non-stationarity, we therefore proceed by inspecting the sample ACF and PACF from the detrended data. These are shown in figure 34.

The cut-off at lag 2 of the PACF and the decay of the ACF would suggest that a possible model for the detrended data would be an AR(2) process (with mean zero). Indeed, this is also the model obtained from `forecast::auto.arima` when choosing the model that minimizes the AICC, after restricting non-significant components to be zero.

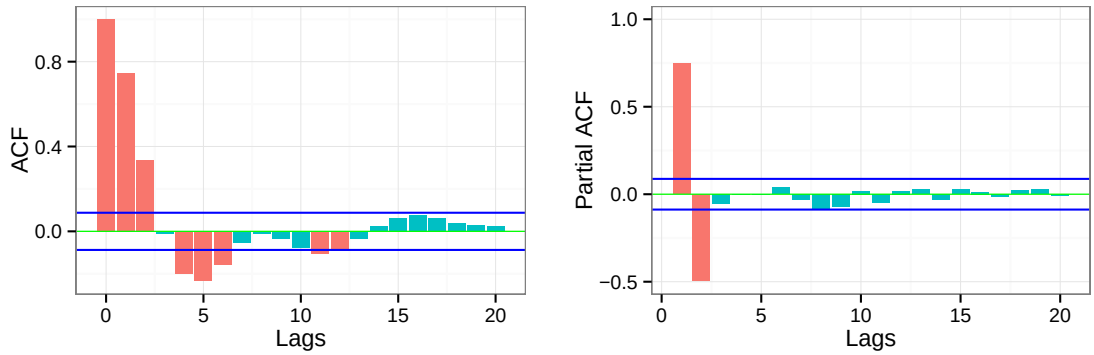


Figure 34: Sample ACF and PACF of detrended data.

We therefore proceed by fitting such a model, again using the Yule-Walker equations and Burg's algorithm. The Yule-Walker equations yield

```
Call:
ar.yw.default(x = d4$stationary, aic = FALSE, order.max = 2,     demean = FALSE)

Coefficients:
     1       2
 1.120  -0.494

Order selected 2  sigma^2 estimated as  0.236
```

while Burg's method yields (apologies for the long line)

```
Call:
ar.burg.default(x = d4$stationary, aic = FALSE, order.max = 2,     demean = FALSE, var.method =

Coefficients:
     1       2
 1.126  -0.502

Order selected 2  sigma^2 estimated as  0.232
```

The test results for the residuals of the AR(2) model fitted by the Yule-Walker equations are shown below. The results for the model fitted by Burg's algorithm are virtually identical, and are therefore omitted.

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)     12.81    0.8853
McLeod-Li Q                Q ~ chisq(20)      15.4    0.7533
Turning points T  (T-330.7)/9.4 ~ N(0,1)       340    0.3203
Diff signs S      (S-248.5)/6.4 ~ N(0,1)       238    0.1035
Rank P         (P-61876.5)/1855 ~ N(0,1)     59759    0.2537
```

None of the test suggest that the residuals are not white noise or iid, giving us confidence that the AR(2) model is a good choice. This is further supported by the diagnostics plots below (which again are almost identical for the Burg model residuals)

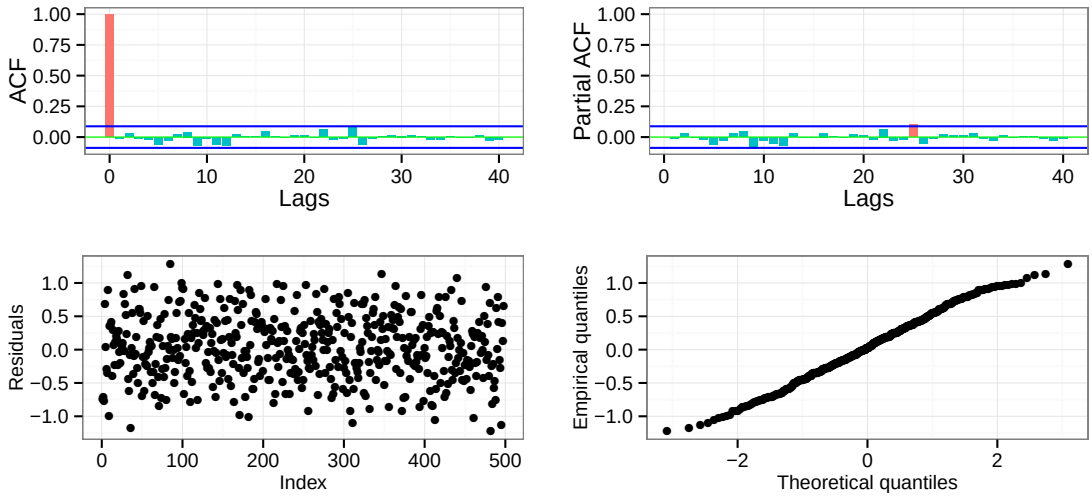


Figure 35: Diagnostics plots for the AR(1) model fitted by the Yule-Walker equations.

Given the similarity of the results between the two estimation methods, we see no reason to prefer one over the other. The final model for the data is

$$
\begin{aligned}
Y_t &= m_t + X_t, \quad \text{with} \\
\hat{m}_t &= 1.2 \cdot 10^{-5} t^2 \\
X_t &= 1.2X_{t-1} - 0.49X_{t-2} + Z_t, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{WN}(0, 0.23)
\end{aligned}
$$

### Results: Dataset 5

The data for this problem is shown in figure 36. As can be seen, dependence is extremely strong, looking very much like a random walk.

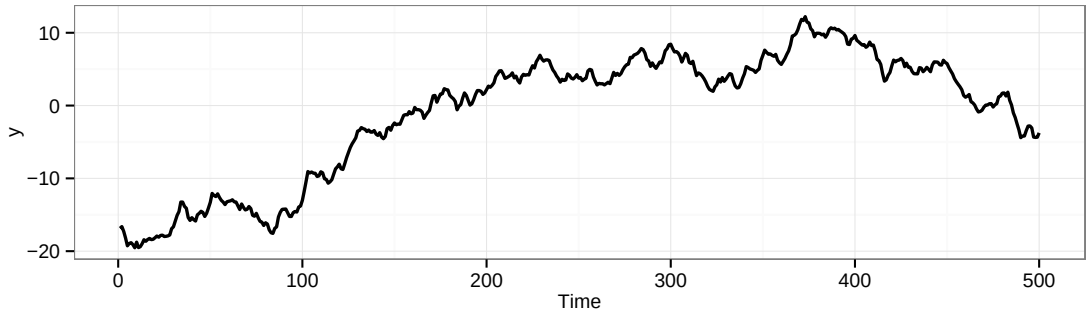


Figure 36: Dataset 5 of Problem 6.

We therefore difference the data (first differences), obtaining the differenced series in figure 37, which looks much more like the realization of a stationary process.

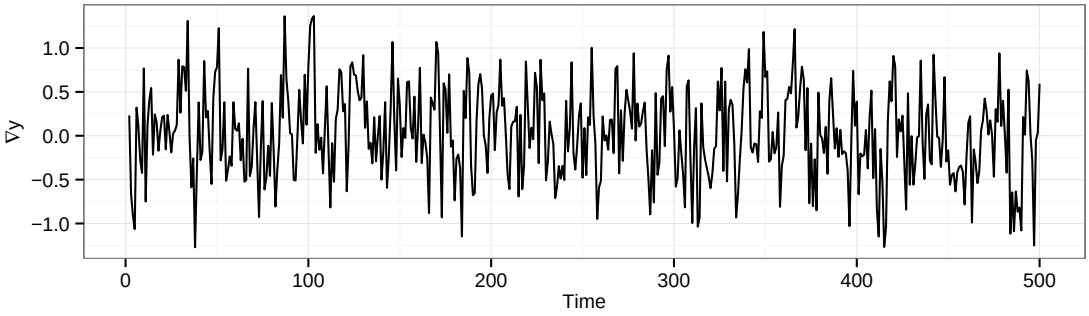


Figure 37: Dataset 5 of Problem 6, after taking first differences.

To see if this is so, we take a look at the sample ACF and and PACF, shown in figure 38.

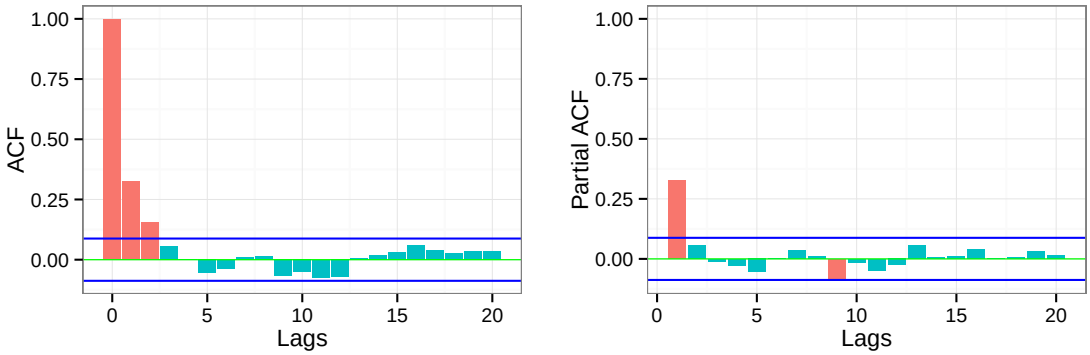


Figure 38: Sample ACF and PACF of differenced data.

Indeed, neither the sample ACF nor PACF show signs of slow decay. Rather, the spike at lag 1 of the PACF and the quick decay of the ACF reminds us of the datasets earlier in Problem 6, when an AR(1) model was found to be appropriate. The simplicity of such a model is alluring, and we therefore fit an AR(1) model (with mean restricted to be zero) to the data to see if the residuals resemble white noise or an iid process. We again use the Yule-Walker equations and Burg's algorithm. For the former, the parameter estimates are

```
Call:
ar.yw.default(x = d5diff$dy, aic = FALSE, order.max = 1, demean = FALSE)

Coefficients:
    1
0.327

Order selected 1  sigma^2 estimated as  0.233
```

while Burg's method yields (apologies for the long line)

```
Call:
ar.burg.default(x = d5diff$dy, aic = FALSE, order.max = 1, demean = FALSE,     var.method = 1)

Coefficients:
    1
0.328

Order selected 1  sigma^2 estimated as  0.232
```

Tests running the same tests as before, first for the model estimated by Burg's algorithm, yields the results

```
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                Q ~ chisq(20)     12.81    0.8855
McLeod-Li Q                Q ~ chisq(20)     14.23    0.8185
Turning points T  (T-330.7)/9.4 ~ N(0,1)       346    0.1026
Diff signs S      (S-248.5)/6.4 ~ N(0,1)       239    0.1407
Rank P         (P-61876.5)/1855 ~ N(0,1)     58493    0.0682
```

This time both the Ljung-Box and McLeod-Li tests suggest that the data (residuals from fitted AR(1) model) is independent, and the other tests also show $p$-values that are sufficiently high for us to believe that the noise is iid. In figure 39 we also show some diagnostics plot that point to the same conclusion:

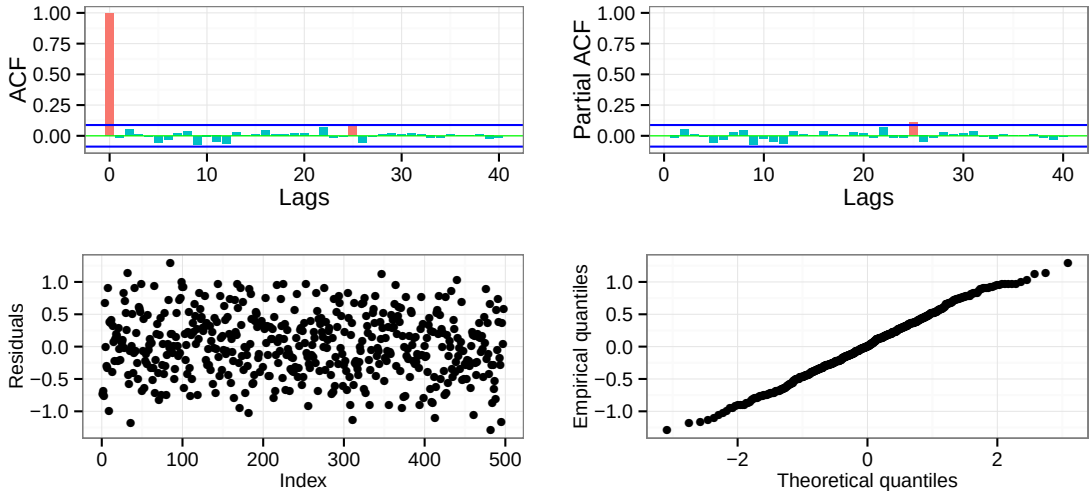


Figure 39: Diagnostics plots for the AR(1) model fitted by Burg's algorithm to the differenced data.

The results for the model fitted by the Yule-Walker equations are essentially identical, and are therefore omitted. However, for the third time in fitting an AR(1) model we note that the estimated noise variance is larger for the YW model than for the Burg model. Beyond that, we see no reason to prefer one approach over the other. The final model for our data is

$$\begin{aligned} Y_t &= Y_{t-1} + X_t, \quad \text{with} \\ X_t &= 0.33 X_{t-1} + Z_t, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{IID}(0, 0.23) \end{aligned}$$

### Results: Dataset 6

In figure 40 we plot the data given for this task. As is, the data already resembles a stationary process, so we continue by examining the sample ACF and PACF if this may be so. On a quick side note, we did try to fit polynomials of order one and two to the data to see if there was any trend that was invisible to the eye, but found that the coefficient estimates were all not significantly different from zero.

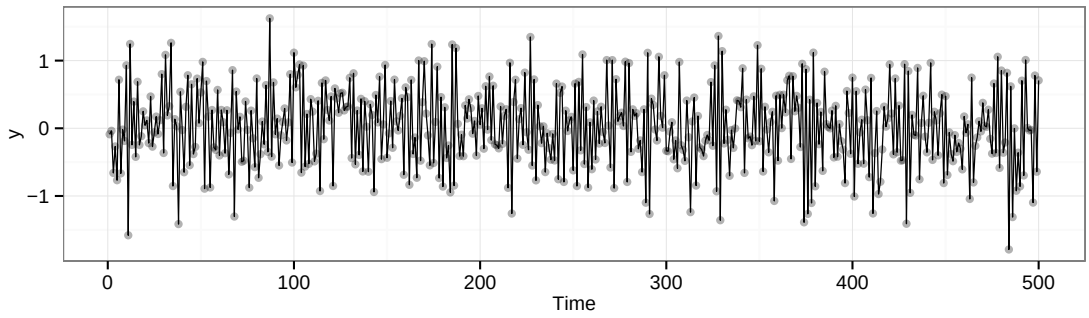


Figure 40: Dataset 6 of Problem 6.

In figure 41, we see spikes at the two first lags of the ACF, and fairly rapidly decaying PACF. Aside from suggesting that the data is stationary, a possible model for the data is that of an MA(2) process. Indeed, this is also the model suggested by comparing AR, MA and ARMA models of different orders and choosing the one AICC, by running `forecast::auto.arima`.

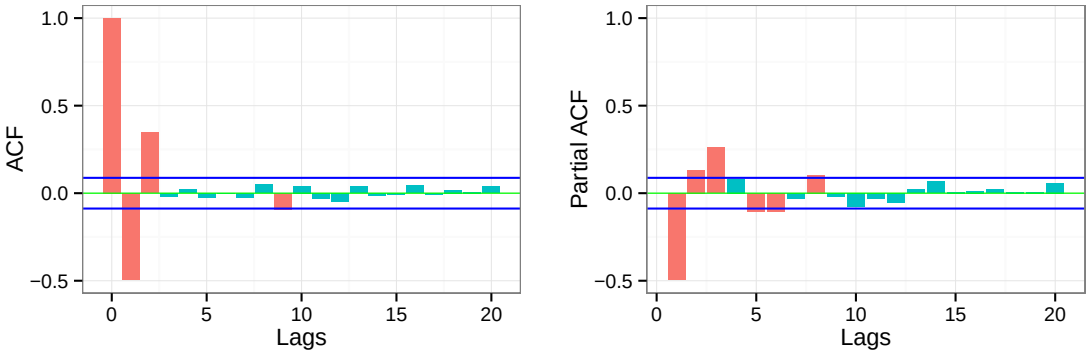


Figure 41: Sample ACF and PACF of differenced data.

We therefore continue by fitting an MA(2) model to the data, using the Innovations algorithm (IA) and the Hannan-Rissanen (HR) algorithm. The IA yields the following parameter estimates:

```
   theta1    theta2    sigma2 se.theta1 se.theta2
   -0.494     0.515     0.229     0.045     0.050
```

while the HR algorithm yields

```
   theta1    theta2    sigma2 se.theta1 se.theta2
   -0.485     0.518     0.229     0.047     0.047
```

As seen, the estimates agree closely, even in the standard errors. The main difference is in the first MA coefficient, which is slightly larger in magnitude for the IA model. Next we run the standard tests on the residuals of these models, to see if they resemble an iid sequence. The results for the IA model are as follows:

```
Null hypothesis: Residuals are iid noise.
Test                         Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)      9.57    0.9754
McLeod-Li Q                 Q ~ chisq(20)     15.37    0.7549
Turning points T    (T-332)/9.4 ~ N(0,1)       335    0.7499
Diff signs S      (S-249.5)/6.5 ~ N(0,1)       245    0.4862
Rank P         (P-62375)/1866.2 ~ N(0,1)     59272    0.0964
```

With high $p$-values for all but the Rank test, these results suggest that the residuals are indeed iid. As with the AR(1) models we saw earlier, the test results and the diagnostics plots for the residuals for the IA and HR models agree very closely, so we omit the result from the HR model here. Figure 42 shows the noise diagnostics plots for the IA model

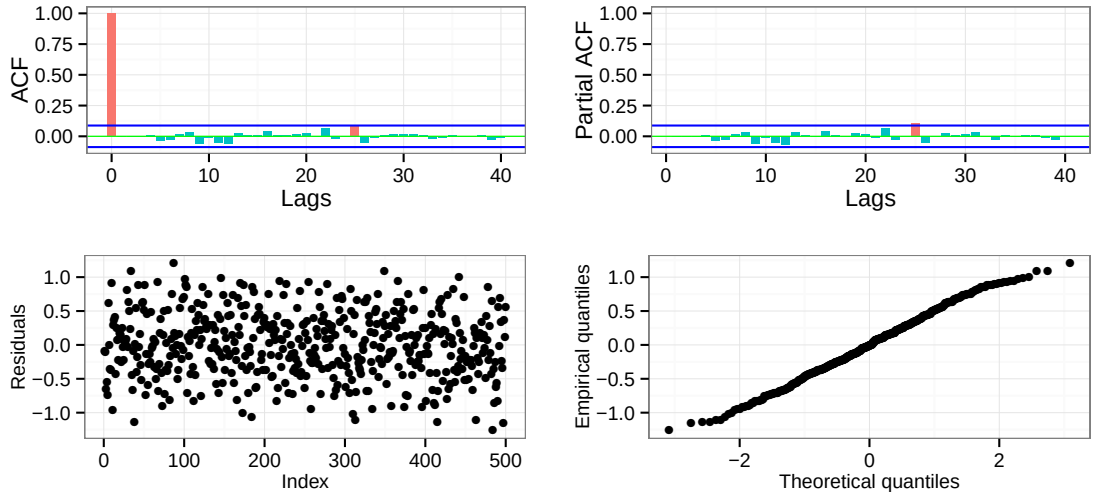


Figure 42: Diagnostics plots for the MA(2) model fitted by the Innovations algorithm.

These plots give no indication that the MA(2) model is a particularly bad choice. We are therefore satisfied with this model for the data, meaning that the final model may be formulated as

$$Y_t = Z_t - 0.49Z_{t-1} + 0.52Z_{t-2}, \quad \{Z_t\} \overset{\text{apx}}{\sim} \text{IID}(0, 0.23)$$

## Problem 7

Here we consider a data set of total energy supply in units GWh for Sweden, obtained from Statistics Sweden (SCB), at http://bit.ly/1mbKfE8. The data also contained supply by source, but we focus on the total here since it would otherwise be more interesting to analyze the supply of the different sources jointly. This particular dataset in interesting from the standpoint of satisfying curiosity, since my knowledge on this subject matter is very limited.

The data is monthly, spanning from January 1974 to February 2014, and is shown in figure 43.

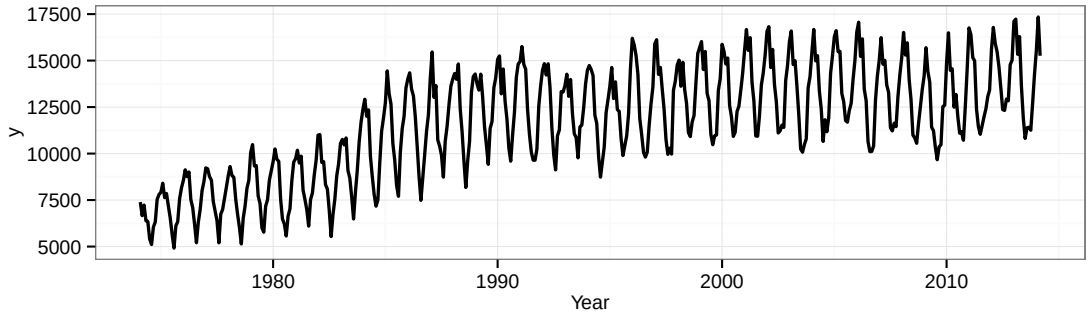


Figure 43: Total monthly energy supply (GWh) for Sweden.

As we can see, there is definitely a cyclical component to the data—energy supply is seen to be greater in the winter months, which is of course natural. We see also that there is a possible trend component to the data, that is increasing at a decreasing rate. This suggests the removal of such a trend component by fitting models in which the power supply is either a square root or logarithm of time, or with a shorter time perspective of how energy supply behaves, a quadratic function with negative coefficient for the quadratic term. Indeed, the latter gives the best result in terms of adjusted $R^2$ and look of the residuals, as shown in figure 44

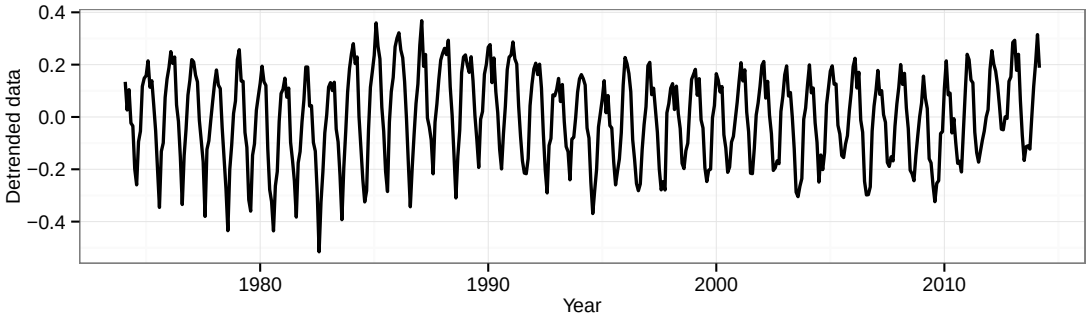


Figure 44

The result of detrending the data is not entirely satisfactory though, and neither were our efforts at removing the season. Fitting and removing a linear combination of sine and cosine terms with period 12 did little to remove the periodicity, and even after using lag-12 differencing and fitting the best possible ARMA model (according to minimum AICC), spikes at multiples of 12 remain in the residual ACF for the fitted ARMA model. We go back to the drawingboard, starting over from

the raw data. We instead apply the method described in section 1.5.2, *Method S1: Estimation of Trend and Seasonal Components* of [1].

To the raw data $\{x_1, \ldots, x_n\}$, we apply a moving average filter

$$\hat{m}_t = (0.5x_{t-6} + x_{t-6+1} + \ldots + x_{t+6-1} + 0.5x_{t+6})/12, \ 6 \leq t \leq n-6$$

in order to eliminate the seasonal component (of period 12) and to dampen noise. For $k = 1, \ldots, 12$, we compute the $w_k$ of the deviations $\{(x_{k+12j} - \hat{m}_{k+12j}), 6 < k + 12j < n - 6\}$. The seasonal component is estimated as

$$\hat{s}_k = w_k - \frac{1}{12}\sum_{i=1}^{1} 2w_i, \quad k = 1, \ldots, 12$$

and $\hat{s}_k = \hat{s}_{k-12}$. We then form the deseasonalized data $d_t = x_t - \hat{s}_t$, $t = 1, \ldots, n$. For the deseasonalized data, we reestimate the trend $\hat{m}_t$, again as a second order polynomial, and finally create the noise series $\hat{Y}_t = x_t - \hat{m}_t - \hat{s}_t$, $t = 1, \ldots, n$. Implemented as described above, the R functions we use are `itsmr::season` and `itsmr::trend`

We plot this data in figure 45.

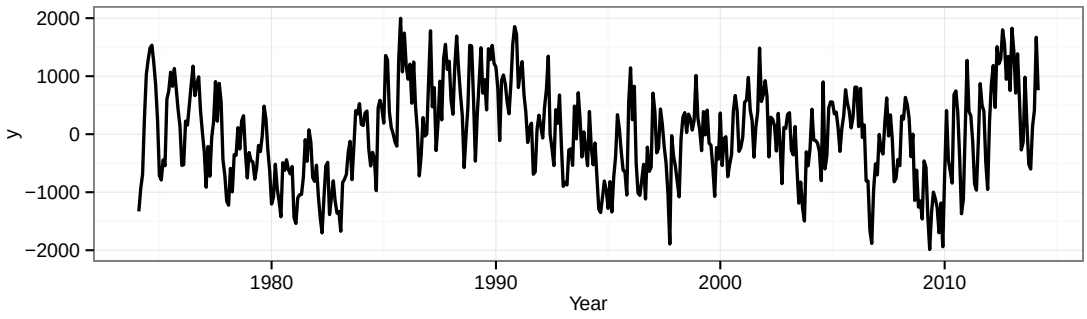


Figure 45: Total monthly energy supply (GWh) after deseasonalizing and detrending.

This data is possibly stationary, although the dependence is clearly strong, as evidenced by the ACF in figure 46 on the next page.

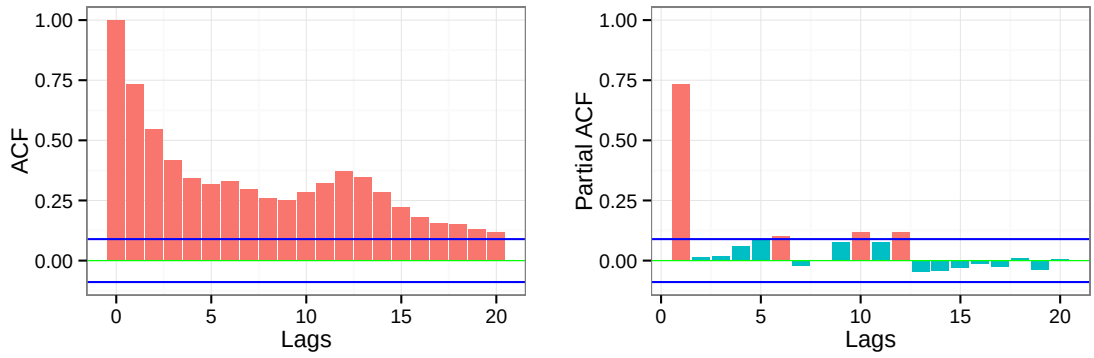


Figure 46: Sample ACF and PACF of the detrended and deseasonalized energy supply data.

To be get an indication that the data is stationary, we perform the Augmented Dickey-Fuller test as described in section 6.3.1 of [1]. The null hypothesis is that there is a unit root; the function `tseries::adf.test` performs this test with the following results:

```
Augmented Dickey-Fuller Test

data:  energy$yt
Dickey-Fuller = -4.76, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

As seen, the null hypothesis of a unit root is rejected, indicating stationarity. We then find the best possible ARMA model by fitting several of them of different orders and choosing the one with lowest AICC. This turns out to be an AR(1) model (which we fit by maximum likelihood), with the following characteristics

```
Series: energy$yt
ARIMA(1,0,0) with zero mean

Coefficients:
        ar1
      0.739
s.e.  0.031

sigma^2 estimated as 289058:  log likelihood=-3715
AIC=7434   AICc=7434   BIC=7442

Training set error measures:
                ME  RMSE   MAE   MPE  MAPE   MASE
Training set 2.063 537.6 437.8 139.9 275.9 0.6681
```

As seen, the AR(1) coefficient is fairly large, and though the noise variance might seem huge, it equates to a standard deviation of 538.

Lastly, we run the same tests as in Problem 6 and 1.

```
Null hypothesis: Residuals are iid noise.
Test                         Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     38.98    0.0067 *
McLeod-Li Q                 Q ~ chisq(20)     26.99    0.1356
Turning points T    (T-320)/9.2 ~ N(0,1)       323    0.7454
Diff signs S      (S-240.5)/6.3 ~ N(0,1)       231    0.1343
Rank P       (P-57960.5)/1766.4 ~ N(0,1)     57447    0.7713
```

We see that the Ljung-Box test rejects the hypothesis of independent data, while the McLeod-Li tests does not. Also inspecting the sample ACF and PACF of the residuals below, we see that our model may not be a perfect fit for the data. Such is life.

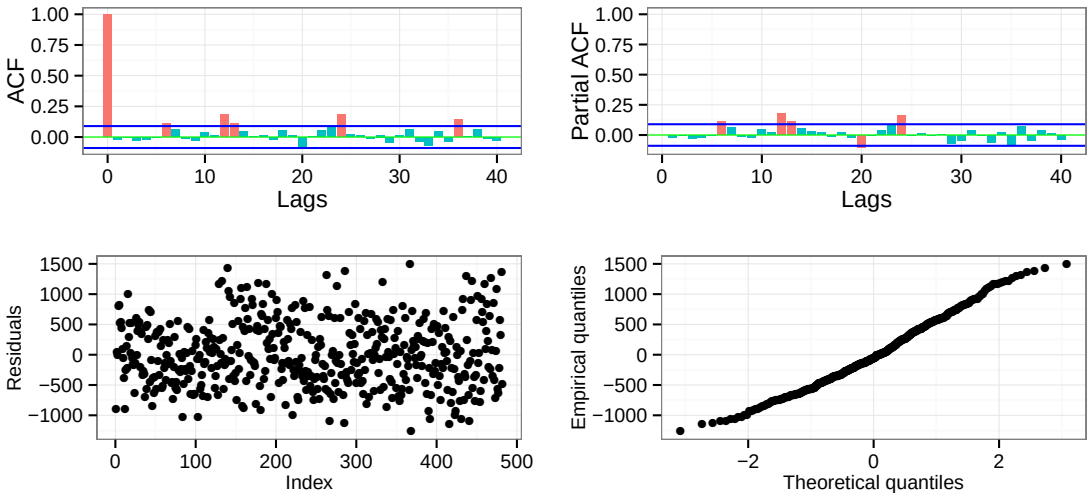


Figure 47: Diagnostics plots for the AR(1) model fitted by maximum likelihood.

## Results

We have looked at a data series consisting of total monthly energy supply for Sweden in the years 1974–2014. This data is clearly seasonal, but initial attempts failed to remove this seasonality. Changing to a non-parametric approach, we improved our luck and arrived at an AR(1) model for the detrended and deseasonalized data. However, the fit of this model is not perfect, but neither should we expect it to be so when dealing with real-world data.

# Problem 8

## Objectives

In this problem we are given data of average monthly atmospheric carbon dioxide levels measured at Mauna Loa, Hawaii, since 1958. More specifically, the file reads that we are given

> ...the monthly mean CO2 mole fraction determined from daily averages. The mole fraction of CO2, expressed as parts per million (ppm) is the number of molecules of CO2 in every one million molecules of dried air (water vapor removed).

Interpolated values are given where data are missing.

Our objective is to predict the average $CO_2$ level for July 2014, with comments on the accuracy of the forecast.

## Methodology and Mathematical Background

Let it first be known that we made several attempts at finding a suitable AR, MA, or ARMA model for the data after transforming the data (including (repeated) differencing) and/or removing fitted trend and seasonal components. The results from these efforts were not satisfactory, so we looked for alternative approaches to the problem. Since the objective is only to forecast, we look to chapter 9 of [1]; particularly the Holt-Winters seasonal algorithm, as described in [1, pp. 326].

The Holt-Winters algorithm when seasonality is not present (or modeled) holds for the case when the data $\{Y_t\}$ are believed to follow the the form $Y_t = m_t + Z_t$, i.e. a trend plus noise form. If the series is stationary and the trend is constant, a natural form of the $h$ step forecast is $P_n Y_{n+h} = \hat{m}_n$. If the trend is non-constant, it seems reasonable to add a first order (slope) component to the forecast, as $P_n Y_{n+h} = \hat{a}_n + \hat{b}_n h$. The one-step forecast is then given by $\hat{Y}_{n+1} := P_n Y_{n+h} = \hat{a}_n + \hat{b}_n$. Then supposing that the estimated level at time $n+1$ will be a linear combination of the observed value and forecasted value at this time, i.e. $\hat{a}_{n+1} = \alpha Y_{n+1} + (1-\alpha)(\hat{a}_n + \hat{b}_n)$, the slope at time $n+1$ may be estimated as $\hat{b}_{n+1} = \beta(\hat{a}_{n+1} - \hat{a}_n) + (1-\beta)\hat{b}_n$. Setting $\hat{a}_2 = Y_2$ and $\hat{b}_2 = Y_2 - Y_1$ as initial conditions, The above recursions may be solved successively. Good values of the smoothing parameters $\alpha$ and $\beta$ can be obtained by minimizing the sum of squares of one-step prediction errors $\sum_{i=3}^{n}(Y_i - P_{i-1}Y_i)^2$ when the algorithm is applied to the given data [1, pp. 322–323]

In the given dataset, we are provided with a trend component, and the seasonality of the average $CO_2$ levels is obvious, having a period of 12. If we therefore believe our series of data $Y_1, \ldots, Y_n$ to consist of a trend and seasonal component with period $d$, the Holt-Winters seasonal algorithm adds to the forecast function above a seasonal component $\hat{c}_{n+h}$, $h > 0$. If $k$ is the smallest integer satisfying $n + h - kd \leq n$, we set $\hat{c}_{n+h} = \hat{c}_{n+h-kd}$, with the values of $\hat{a}_n, \hat{b}_n, \hat{c}_n$ obtained from the recursions

$$\begin{aligned}
\hat{a}_{n+1} &= \alpha(Y_{n+1} - \hat{c}_{n+1-d}) + (1-\alpha)(\hat{a}_n + \hat{b}_n) \\
\hat{b}_{n+1} &= \beta(\hat{a}_{n+1} - \hat{a}_n) + (1-\beta)\hat{b}_n \\
\hat{b}_{n+1} &= \gamma(Y_{n+1} - \hat{a}_{n+1}) + (1-\gamma)\hat{c}_{n+1-d}
\end{aligned}$$

with initial conditions $\hat{a}_{d+1} = Y_{d+1}$, $\hat{b}_{d+1} = (Y_{d+1} - Y_1)$, and $\hat{c}_i = Y_i - (Y_1 + \hat{b}_{d+1}(i-1))$, $i = 1 \ldots, d+1$. Again optimal values for $\alpha$, $\beta$ and $\gamma$ can be obtained by minimizing the sum of squared one-step errors $\sum_{i=d+2}^{n}(Y_i - P_{i-1}Y_i)^2$.

## Results

The given data is shown in figure 48 with the monthly average (interpolated) data and the trend component provided.

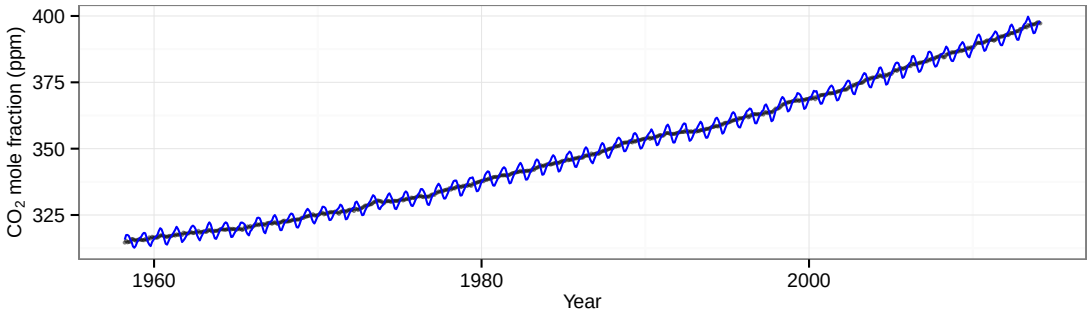


Figure 48: Atmospheric carbon dioxide levels at Manua Loa, Hawaii.

The R function `stats::HoltWinters` allows for Holt-Winters filtering of data using the seasonal algorithm discussed above, with parameters chosen to minimize the squared prediction error. The specific value that these estimated parameters and model components take is of lesser interest here; instead we are interested in how reasonable the forecast seems. In figure 49 we show the (up to) 5-step forecasts using the available data, which is up to February 2014. The 5-step forecast gives us the estimate for July 2014.

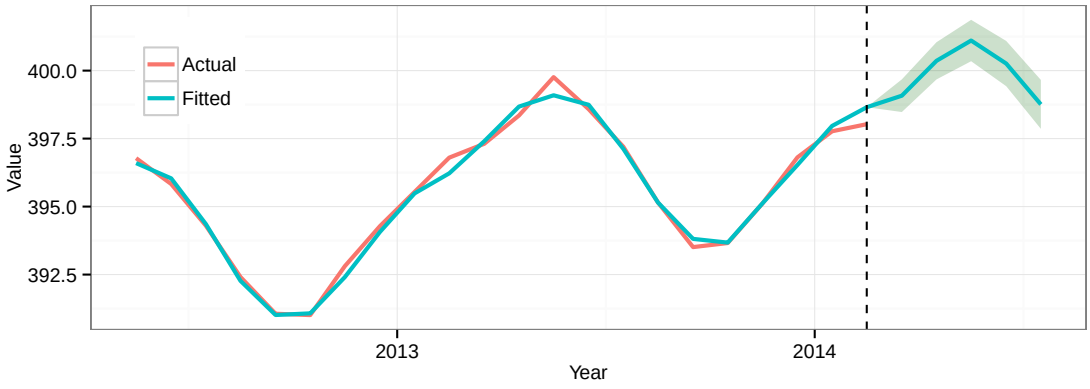


Figure 49: Forecast of monthly mean carbon dioxide levels at Manua Loa, Hawaii, up to July 2014, using data up to and including February of the same year.

The mean forecast for the average $CO_2$ level in July 2014 is 398.76 ppm, with a 95% prediction interval of (397.9, 399.7). This interval is quite narrow, suggesting high confidence in the forecast. We also know from the look of the data that the $CO_2$ levels in one particular month and year will be very similar to those precisely one year earlier, but a bit higher. Only time will tell how accurate the forecast really was.

## References


[1] Brockwell, Peter J., and Davis, Richard A., *Introduction to Time Series and Forecasting*. Springer, 2nd edition, 8th corrected printing, 2010.

[2] Ruey S. Tsay, *Analysis of Financial Time Series*. Wiley, 3rd edition, 1st printing, 2010.

[3] Wikipedia, *Komolgorov-Smirnov test*, http://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test.

[4] R Documentation, *Fit Autoregressive Models to Time Series*, http://stat.ethz.ch/R-manual/R-patched/library/stats/html/ar.html.

[5] Donald B. Percival, *Diagnostics tests*, University of Washington, http://faculty.washington.edu/dbp/s519/R-code/diagnostic-tests.R.

[6] Wikipedia, *Akaike information criterion*, http://en.wikipedia.org/wiki/Akaike_information_criterion.

[7] Hadley Wickham, *ggplot2: elegant graphics for data analysis*, Springer, 2009, http://had.co.nz/ggplot2/book.