10 | Time Series Models

Time series is a sequence of observations over time. To analyze a very general class of time series will be impossible. Some assumptions have to be made. One such assumption is stationarity.

Stationary Process

Strict stationarity A time series $\{Y_t\}$ is said to be strictly stationary if the joint distributions of $(Y_{t_1}, \dots, Y_{t_k})^T$ and $(Y_{t_1+h}, \dots, Y_{t_k+h})^T$ are the same for all positive k and for all $t_1, \dots, t_k, h \in \mathbb{Z}$.

Strict stationarity means that the probabilistic behavior of any subset of realizations is identical to that of the shifted set. Intuitively the graphs over two equal-length time intervals of realizations of the time series should exhibit similar statistical characteristics.

Strict stationarity is too strong a condition for most applications. It is common to assume that a time series is weakly stationarity.

Weak stationarity A time series $\{Y_t\}$ is said to be weakly stationary or simply stationary if

(i) $E(Y_t) = \mu < \infty$, $\forall\, t \in \mathbb{Z}$
(ii) $\text{Cov}(Y_s, Y_t) = \text{Cov}(Y_{s+r}, Y_{t+r}) = \gamma_{s-t} < \infty$, $\forall\, r, s, t \in \mathbb{Z}$

We will use the weak stationarity definition for a process being stationary unless it is specified otherwise.

Weak stationarity is also called covariance stationarity or second order stationarity, because the conditions only on the first two moments of Y_t. The covariance among time series $\text{Cov}(Y_s, Y_t)$ is called *autocovariance*. A weakly stationary time series has constant mean, constant variance and its autocovariance is a function only of time separation or lag h.

Eg 10.1. White noise. A sequence $\{\varepsilon_t\}$ with $E(\varepsilon_t) = 0$, $\forall t$,

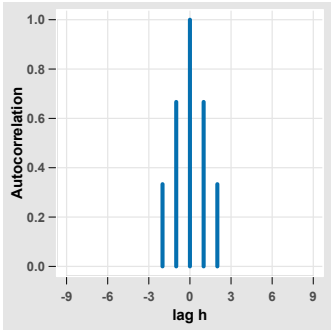
$$\text{Cov}(\varepsilon_s, \varepsilon_t) = \begin{cases} \sigma^2 & , \ s = t \\ 0 & , \ s \neq t \end{cases} \implies \gamma_h = \begin{cases} \sigma^2 & , \ h = 0 \\ 0 & , \ h \neq 0 \end{cases}$$

Eg 10.2. Moving average. Let $\varepsilon_t \sim WN(0,1)$ and

$$Y_t = \frac{1}{3}(\varepsilon_{t-1} + \varepsilon_t + \varepsilon_{t+1})$$

Then the autocovariance at lag h is

$$\gamma_h = \begin{cases} 3/9 & , \ h = 0 \\ 2/9 & . \ h = \pm 1 \\ 1/9 & , \ h = \pm 2 \\ 0 & , \ |h| \geq 3 . \end{cases}$$



The Autocorrelation function, ACF

The autocorrelation of a time series $\{Y_t\}$ is the correlation between Y_s and Y_t: If $\{Y_t\}$ is stationary then the autocorrelation function

$$\rho_h = \frac{\gamma_h}{\gamma_0} .$$

The sample aucovariances and autocorrelations are

$$\hat{\gamma}_h = \frac{1}{n}\sum_{t=1}^{n-h}(Y_{t+h} - \bar{Y})(Y_t - \bar{Y}) = \frac{1}{n}\sum_{t=h+1}^{n}(Y_t - \bar{Y})(Y_{t-h} - \bar{Y}),$$

$$\hat{\rho}_h = \frac{\hat{\gamma}_h}{\hat{\gamma}_0} .$$

For large n, the sample autocorrelation $\hat{\rho}_h$ is approximately normal with mean ρ_h and variance depending on the true ρ_h.

ACF of Independent White Noise If $\{Y_t\} \sim i.i.d.(0, \sigma^2)$, then for large n, $\hat{\boldsymbol{\rho}}_\ell = (\hat{\rho}_1 \ldots, \hat{\rho}_\ell)^T$ with ℓ fixed

$$\hat{\boldsymbol{\rho}}_\ell \ \text{ is asymptotically } \ N(\mathbf{0}, n^{-1}\mathbf{I}_\ell) . \qquad (10.1)$$

This result can be used to test for the null $H_0 : \rho_h = 0$ against $H_A : \rho_h \neq 0$ by comparing $\sqrt{n}\,\hat{\rho}_h$ with $z_{\alpha/2}$ from the standard normal. More importantly, the joint distribution result of (10.1) is used to derive a portmanteau statistic which tests several autocorrelations are zero simultaneously.

Portmanteau Tests for White Noise

A portmanteau test takes into consideration the magnitudes of $\hat{\rho}_h$ as a group. Box and Pierce (1970) propose the Portmanteau statistic $n\sum_{\ell=1}^{K}\hat{\rho}_\ell^2$ as a test statistic for the null hypothesis,

$$H_0 : \rho_1 = \cdots = \rho_K = 0 .$$

Under the null hypothesis, the Box-Pierce statistic has an asymptotic χ^2 distribution with K degrees of freedom by (10.1). The statistic in

small samples approximates the χ^2 distribution poorly. Ljung and Box modify the statistic to correct the bias and improve the power of the test in finite samples,

$$Q(K) = n(n+2)\sum_{\ell=1}^{K} \frac{\hat{\rho}_\ell^2}{n-\ell} \tag{10.2}$$

The statistic is a weighted sum of squred sample ACF over the first K lags, $Q(K)$ approximates a χ^2_K distribution better under the null. The R function for the Ljung and Box test is `Box.test()`.

Eg 10.3. The data in this example are daily returns in % of Bristol-Myers Squibb (BMY), Kellanova (K, formally Kellogg) and Eastman Kodak (KODK), from January 1, 2015 to October 31, 2024. Sample size $n = 2475$.

```
head(Y3,2); tail(Y3,2); n = dim(Y3)[1]; cat("\nsample size:", n)
```

```
##                   BMY           K      KODK
## 2015-01-02  0.8098569  0.06110116  1.644626
## 2015-01-05 -1.5750965 -1.88073732 -3.269916
##                   BMY           K      KODK
## 2024-10-30 0.3042982  0.3469199 -2.874939
## 2024-10-31 5.7379947 -0.2476744 -1.680720
##
## sample size: 2475
```

Figure10.1 shows the autocorrelation plots of the three asset daily returns. The reference band in the ACF plot is between $\pm 1.96/\sqrt{n}$ which are the test bounds for testing the null hypothesis that an autocorrelation is 0 at the 0.05 level. Unlike the Portmanteau statistic $Q(K)$, these tests test autocorrelations one at a time. Both Bristol-

Myers and Kellogg show little evidence of serial correlation, while Kodak has notable ACF at lags 1 and 3.

The R function that calculates and plots the sample ACF is `acf()`. The sample ACF plot however starts with lag 0, that is, $\hat{\rho}_0 = 1$. The `forecast` package has an R function `Acf()` which does everything as `acf()` but starts with the 1st lag.

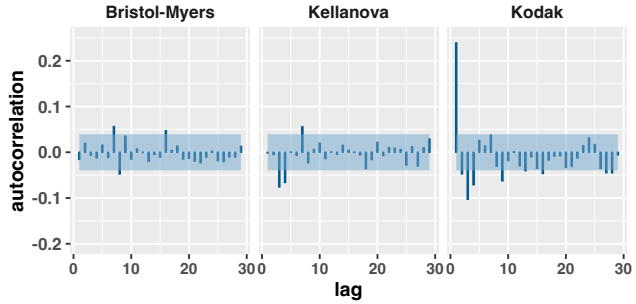


Figure 10.1: The ACF of daily returns of 3 assets in Eg. 10.3

The function `Box.test()` has Box-Pierce statistic as default, we should change it to Ljung-Box by setting `type = "L"`

```
args(Box.test)
```

```
## function (x, lag = 1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
```

```
## Ljung-Box test for Kellogg daily returns with K = 4
out =Box.test(Y3[,"K"], 4, type = "L"); out
```

```
##  Box-Ljung test
##
## data:  Y3[, "K"]
## X-squared = 25.233, df = 4, p-value = 4.517e-05
```

```
names(out)
```

```
## [1] "statistic" "parameter" "p.value"   "method"    "data.name"
```

In practice, the selection of K will affect the performance of the $Q(K)$ statistic. We may want to use a small K, because serial correlation is often the strongest at lower lags. This also avoids large estimating errors of $\hat{\rho}_h$ at higher lags in finite sample due to its calculation being a normalized sum of $n-h$ terms. However, using a too small value of K may miss serial correlation beyond lag K. Simulation studies suggest that the choice of $K \approx \log(n)$ provides better power performance. We suggest $K = \log n$ to be smallest lag to examine. It is not uncommon to have tests with several values of K to check if a process is white noise. In this example, $\log n = \log 804 = 7.81 \approx 8$.

```
## p-values of Ljung-Box tests for White Noise
Ks = 2 + 3*(0:5); pv = c()
for(i in Ks) pv = cbind(pv,apply(Y3, 2, function(x) Box.test(x, i, "L")$p.val))
colnames(pv) = paste("K = ", Ks); round(pv,5)

##       K =  2  K =  5  K =  8 K =  11 K =  14 K =  17
## BMY  0.46058 0.75700 0.03686 0.04279 0.09631 0.05736
## K    0.96698 0.00013 0.00004 0.00018 0.00088 0.00378
## KODK 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
```

Clearly Kodak returns are not white noise, Bristol -Myers returns are white noise. Kellogg returns may have some weak evidence of serial correlation. We will discuss the last case lated.

Autoregressive Models

Autoregressive models are created with the idea that the present value of the series Y_t, can be explained as a linear function of p past values of $Y_{t-1}, \ldots, Y_{t-p}$. An autoregressive process $\{Y_t, t = 0, \pm 1, \pm 2, \ldots\}$ of order p, abbreviated as AR(p) satisfy the linear

difference equation,

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t, \qquad \varepsilon_t \sim WN(0, \sigma_\varepsilon^2), \quad (10.3)$$

where $\phi_1, \ldots, \phi_p$ are constants and $c = \mu(1 - \phi_1 - \cdots - \phi_p)$ thus $\mu = E(Y_t)$. Define the backshift operator B by

$$BY_t = Y_{t-1} \quad \text{and} \quad B^k Y_t = Y_{t-k}$$

Then, the AR(p) model can be written either as

$$(1 - \phi_1 B - \cdots - \phi_p B^p) Y_t = c + \varepsilon_t$$
$$\text{or} \quad (1 - \phi_1 B - \cdots - \phi_p B^p)(Y_t - \mu) = \varepsilon_t,$$

Define further the autoregressive operator $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$, then the AR($p$) is

$$\phi(B)(Y_t - \mu) = \varepsilon_t.$$

For simplicity in notation, we assume $\mu = 0$ and thus $\phi(B) Y_t = \varepsilon_t$ in our discussion. If $\mu \neq 0$, we can always replace Y_t by $Y_t - \mu$. We initiate the study of AR models by the first order model, AR(1).

The AR(1) Models

An AR(1) model is given by $Y_t = \phi Y_{t-1} + \varepsilon_t$. Iterated backwards m times, we get

$$Y_t = \phi^m Y_{t-m} + \sum_{j=0}^{m-1} \phi^j \varepsilon_{t-j}. \quad (10.4)$$

Providing $|\phi| < 1$.

By continuing to iterate backwards, the AR(1) model has an infinite sum representation of the current and past innovations (or shocks),

$$Y_t = \sum_{j=0}^{\infty} \phi^j \varepsilon_{t-j}\,, \tag{10.5}$$

An AR(1) is stationary if $|\phi| < 1$, in such cases, its variance and autocorrelations are

$$\text{var}(Y_t) = \gamma_0 = \frac{\sigma_\varepsilon^2}{1-\phi^2}\,, \quad \rho_h = \phi^{|h|}, \quad \forall\, h, t\,. \tag{10.6}$$

The autocorrelation function satisfies the recursion

$$\rho_h = \phi\rho_{h-1}\,. \qquad h \geq 1\,.$$

We will discuss the ACF of a general AR(p) model later.

The left panel of Figure 10.2 shows the autocorrelations of two AR(1) models with $\phi = 0.8$ and -0.8 repectively. Both plots show the ACF decays rapidly. For a negative ϕ, the ACF alternates.

Convergence to the Stationary Distribution Suppose that Y_0 is an arbitrary starting value not chosen from the stationary distribution and that $Y_t = \phi Y_{t-1} + \varepsilon_t$ holds for $t = 1, 2, \ldots,$. Then the process is not stationary, but converges to the stationary distribution satisfying (10.6) . The convergence to the stationary distribution can be very rapid which $|\phi|$ is not too close to 1.

Nonstationary AR(1) processes If $|\phi| \geq 1$, then the AR (1) process will not satisfy the conditions of weak stationarity or cannot be represented by a function of the past observations.

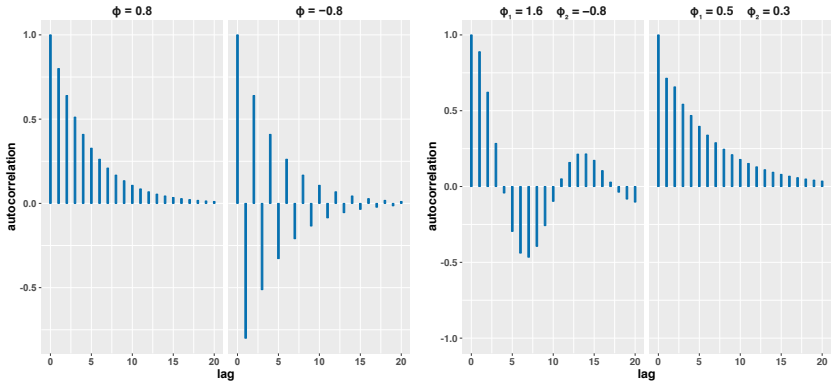


Figure 10.2: The left panel are the autocorrelations of two AR(1) models with $\phi = 0.8$ and -0.8. The right are the autocorrelations of two AR(2) models with coefficients $(\phi_1, \phi_2)^T = c(1.6, -0.8)^T$ and $(\phi_1, \phi_2)^T = c(0.5, 0.3)^T$.

Random Walk. If $\phi = 1$, then $Y_t = Y_{t-1} + \varepsilon_t$. This is the random walk process. Suppose $Y_0 = 0$, then $\text{Cov}(Y_t, Y_{t+h}) = t\sigma_\varepsilon^2$, $h \geq 0$, implying $\rho_h = 1, h \geq 0$. The variance which depends on t is not a constant. Clearly, Random Walk is not stationary.

Explosive Process. If $|\phi| > 1$, the AR(1) is an explosive process, from (10.4), we see that Y_t depends more strongly on the distant past than on the recent past. However, we can obtain stationarity by writing

$$Y_{t+1} = \phi Y_t + \varepsilon_{t+1} \qquad \text{and} \qquad Y_t = \phi^{-1} Y_{t+1} - \phi^{-1}\varepsilon_{t+1}.$$

By iterating forward m times, we would get

$$Y_t = \phi^{-m} Y_{t+m} - \sum_{j=1}^{m-1} \phi^{-j}\varepsilon_{t+j}\,.$$

Because $|\phi| > 1$, thus $|\phi^{-1}| < 1$. This leads to the stationary future dependent AR(1) process,

$$Y_t = -\sum_{j=1}^{\infty} \phi^{-j} \varepsilon_{t+j}. \tag{10.7}$$

Unfortunately, this model is not useful because it requires us to know the future to be able to predict the future.

Causality An AR(1) process is causal if it has the infinite sum representation as (10.5) that does not depend on the future shocks. When $|\phi| > 1$, the AR(1) is a non causal process as shown in (10.7). An AR(1) process is stationary and causal if and only if $\phi < 1$.

Estimation of AR(1) Processes

R has the function `arima()` for fitting AR and more general ARMA time series models. The function `arima()` has two primary estimation methods, conditional least-squares and maximum likelihood. They are similar and generally give nearly the same estimates. The default method in R's `arima()` is the MLE with the conditional least-squares estimate as the starting value for computing the MLE with iterative nonlinear optimization.

Maximum likelihood estimation Let $\boldsymbol{Y} = (Y_1, \ldots Y_t)^T$ be taken from a stationary AR(1) process, then the joint PDF of $\boldsymbol{Y}$ can be written as a product of conditional densities,

$$f_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n) = f_{Y_1}(y_1) f_{Y_2|Y_1}(y_2|y_1) \cdots f_{Y_n|Y_{n-1}}(y_n|y_{n-1}).$$

The Gaussian likelihood of AR(1) model is based on the densities of the following distribution and conditional distribution,

$$Y_1 \sim N\left(\mu, \frac{\sigma_\varepsilon^2}{1-\phi^2}\right),$$

$$Y_t|Y_{t-1} \sim N\left(\mu + \phi(Y_{t-1}-\mu), \sigma_\varepsilon^2\right), \quad t \geq 2.$$

The log-likelihood of $\boldsymbol{Y}$ is

$$\log L(\mu, \phi, \sigma_\varepsilon^2|\boldsymbol{Y}) = -\frac{n}{2}\log\ \sigma_\varepsilon^2 + \frac{1}{2}\log(1-\phi^2) - \frac{1}{2\sigma_\varepsilon^2} S(\mu,\phi) + \text{Constant},$$

where the *unconditional sum of squares*,

$$S(\mu,\phi) = (1-\phi^2)(Y_1-\mu)^2 + \sum_{t=2}^{n}\left\{(Y_t-\mu) - \phi(Y_{t-1}-\mu)\right\}^2.$$

For each value of $(\mu,\phi)^T$, the MLE of σ_ε^2 is $n^{-1}S(\mu,\phi)$, thus we can instead work with the profile likelihood to simplify the objective function of our optimization problem.

The profile likelihood of $(\mu,\phi)^T$ is

$$\log L_p(\mu,\phi) = -\frac{n}{2}\log\left\{\frac{1}{n}S(\mu,\phi)\right\} + \frac{1}{2}\log(1-\phi^2).$$

The MLE $(\hat{\mu},\hat{\phi})$ is obtained by maximizing the log profile likelihood over (μ,ϕ) and $\hat{\sigma}_\varepsilon^2 = n^{-1}S(\hat{\mu},\hat{\phi})$.

Conditional least squares estimator Conditioning on the initial value, Y_1, the conditional likelihood becomes

$$\log L(\mu,\phi,\sigma_\varepsilon^2|y_1) = -\frac{n-1}{2}\log\sigma_\varepsilon^2 - \frac{1}{\sigma_\varepsilon^2}S_C(\mu,\phi)$$

where the conditional sum of squares

$$S_C(\mu,\phi) = \sum_{t=2}^{n}\left\{(Y_t-\mu)-\phi(Y_{t-1}-\mu)\right\}^2.$$

The conditional MLE of σ_ε^2 for a fixed value of (μ,ϕ) is $(n-1)^{-1}S_C(\mu,\phi)$, the log conditional profile likelihood of $(\mu,\phi)^T$

$$\log L_p(\mu,\phi|Y_1) = -\frac{n-1}{2}\log S_C(\mu,\phi).$$

Conditioning on Y_1, the AR(1) model is simply a linear model that can be estimated by the least squares regression method. That is, the conditional MLE of $(\mu,\phi)^T$ is the least squares estimators obtained by minimizing

$$\sum_{t=2}^{n}\left\{Y_t-(c+\phi Y_{t-1})\right\}^2,$$

where $c=\mu(1-\phi)$ is the intercept.

Most of concepts discussed for AR(1) generalized easily to the higher order autoregressive model, the AR(p) models.

The AR(p) Models

We see immediately that the conditional least squares estimator of an AR(p) is simply the multiple regression of Y_t on $Y_{t-1},\dots Y_{t-p}$.

However, the analogous condition to $|\phi|<1$ of AR(1) requires more details. The condition for an AR(p) to be stationary and causal is that all the roots of the polynomial equation associated to the autoregressive operator,

$$\mathscr{P}_\phi(z) = z^p - \phi_1 z^{p-1} - \cdots - \phi_{p-1}z - \phi_p, \tag{10.8}$$

lie inside the unit circle in the complex plane.

Remark 10.1. Some textbooks write the AR polynomial as $\phi(z)$, plugging z to the AR operator. The resulting polynomial is

$$\mathscr{P}_\phi^\dagger(z) = 1 - \phi_1 z - \cdots - \phi_p z^p, \tag{10.9}$$

The roots of (10.9) are reciprocal of those of (10.8). That is, if z_0 is such that $\mathscr{P}_\phi(z_0)=0$, then $\mathscr{P}_\phi^\dagger(1/z_0)=0$. Thus, the equivalent condition of stationarity in terms of polynomial (10.9) is that its roots lie outside of the unit circle in the complex plane.

The ACF of AR(p) are satisfy the following equations,

$$\begin{aligned}\gamma_0 &= \phi_1\gamma_1+\cdots+\phi_p\gamma_p+\sigma_\varepsilon^2\\ \gamma_h &= \phi_1\gamma_{h-1}+\cdots+\phi_p\gamma_{h-p}, \qquad h\geq 1.\end{aligned} \tag{10.10}$$

The first $p+1$ equations are called Yule-Walker equations.

The ACFs can be computed using R's `ARMAacf()` function.

```
gamma.1 = ARMAacf(ar=c(1.6,-0.8),lag.max=20)
gamma.2 = ARMAacf(ar=c(0.50, 0.3),lag.max=20)
```

The right panel of Figure 10.2 is the true ACF plots of two AR(2) models. When the roots of $\mathscr{P}_\phi(z)$ are complex, the process exhibits pseudo-cyclic behavior as we can see from the ACF plot.

Eg 10.4. Fitting an AR(p) model. The data in this example is the weekly returns of Kodak from Eg. 10.3. First R function to be used is `arima()`.

```
args(arima)

## function (x, order = c(0L, 0L, 0L), seasonal = list(order = c(0L,
##     0L, 0L), period = NA), xreg = NULL, include.mean = TRUE,
##     transform.pars = TRUE, fixed = NULL, init = NULL, method = c("CSS-ML",
##         "ML", "CSS"), n.cond, SSinit = c("Gardner1980", "Rossignol2011"),
##     optim.method = "BFGS", optim.control = list(), kappa = 1e+06)
```

The argument `order` is where the order p being specified, it takes `c(p,d,q)`. For an AR model, `order = c(p,0,0)`. For example, we want to fit an AR(3) model to the KODK returns.

```
yt = Y3[,"KODK"]
fit.ar3 = arima(yt, order = c(3,0,0)); fit.ar3

## Call:
## arima(x = yt, order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3  intercept
##       0.2578  -0.0925  -0.0685    -0.0607
## s.e.  0.0200   0.0206   0.0201     0.1297
##
## sigma^2 estimated as 33.97:  log likelihood = -7874.71,  aic = 15759.42
```

The name `intercept` in the above output is somewhat confusing, it is the estimate of the mean $\mu = E(Y_t)$ instead of the "intercept"

defined in (10.3), where the intercept $c = \mu(1-\phi_1-\phi_2-\phi_3)$. R's `arima()` will fit the mean by default.

R's `arima()` is meant for fitting a very general class of ARMA types of models that can also accommodate trends and seasonalities. For fitting pure autoregressive models, the `ar()` function is an alternative R function to use. There are four estimation methods to choose in `ar()`, the Burg, the Yule-Walker, the OLS and the MLE. The default method is the Yule-Walker estimation.

```
args(ar)

## function (x, aic = TRUE, order.max = NULL, method = c("yule-walker",
##     "burg", "ols", "mle", "yw"), na.action = na.fail,
##     series = deparse(substitute(x)), ...)
```

We can select an estimation by setting `method` in `ar()` or use `ar` with an extension, `ar.mle()`, `ar.ols()`, etc. The Yule-Walker and OLS estimations can be applied to fit multivariate time series. The default fitting will select the order p with the AIC criterion. The function can be used for fitting a specific order p by setting `aic = F` and `order.max = p`. We show the OLS estimation with order selection by the AIC.

```
ar.ols(as.vector(yt), order.max = 6)
## Call:
## ar.ols(x = yt, order.max = 6)
##
## Coefficients:
##       1        2        3        4        5
##  0.2569  -0.0930  -0.0554  -0.0461   0.0431
##
## Intercept: 0.002151 (0.1172)
##
## Order selected 5  sigma^2 estimated as  33.91
```

Moving Average Processes and the ARMA Models

We may think of ε_t as "shock" to the process at time t and that what happens today might be related to shocks from a few previous days, $\varepsilon_{t-1}, \ldots, \varepsilon_{t-q}$. We can model our data that depend only on a finite number of the past shocks, such a model is called the moving average model. We can also consider a model depending on both past observations and shocks, this class of models is call the autoregressive moving average models, or simply the ARMA models.

The Moving Average Models

The moving average model of order q, or the MA(q) model for a process $\{Y_t\}_{t=\infty}^{\infty}$, is defined as

$$Y_t - \mu = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}, \qquad \varepsilon_t \sim WN(0, \sigma_\varepsilon^2). \quad (10.11)$$

As in the AR(p), with the backshift operator B and defining a moving average operator in B, we can write the MA(q) as

$$Y_t - \mu = (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t = \theta(B)\varepsilon_t .$$

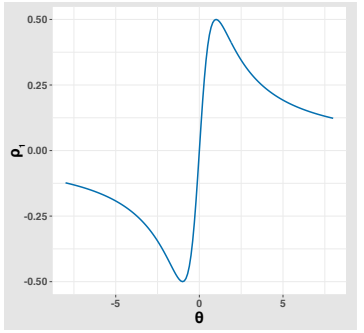
Unlike the autoregressive process, the moving average process is stationary for any values of the parameters $\theta_1, \ldots, \theta_q$.

The MA(1) Process The MA(1) model, $Y_t = \varepsilon_t + \theta \varepsilon_{t-1}$ has autocovariances,

$$\gamma_h = \begin{cases} (1+\theta^2)\sigma_\varepsilon^2 & , \ h = 0 \\ \theta \sigma_\varepsilon^2 & , \ h = 1 \\ 0 & , \ h > 1 . \end{cases}$$

The autocorrelation ACF is

$$\rho_h = \begin{cases} \dfrac{\theta}{1+\theta^2} & , \ h = 1 \\ 0 & , \ h > 1 . \end{cases}$$



Note that $|\rho_1| \le 1/2$ for all values of θ. Unlike AR(1), Y_t is correlated with Y_{t-1}, but not with $Y_{t-2}, Y_{t-3} \ldots$.

We see ρ_1 is the same for θ and $1/\theta$. The model with $(\sigma_\varepsilon^2, \theta) = (1, 5)$ has the same autocovariance function as that of the model of $(\sigma_\varepsilon^2, \theta) = (25, 1/5)$. Thus, MA models defined in (10.11) without any conditions are not unique. We will need to choose only one of the two models with identical autocovariance function.

Invertibility For the sake of model identifiability, we will choose the model such that ε_t is expressible in terms of present and past values of the process Y_t, $s \le t$. Such a process is called an invertible process. Write the MA(1) as $\varepsilon_t = -\theta \varepsilon_{t-1} + Y_t$. Following the steps that led to (10.5), if $|\theta| < 1$, then

$$\varepsilon_t = \sum_{j=0}^{\infty} (-\theta)^j Y_{t-j} .$$

MA(q) Process It is easy to see the ACF of an MA(q) is zero beyond order q. The condition of invertibility for an MA(q) model is that all the roots of the polynomial

$$\mathscr{P}_\theta(z) = z^q + \theta_1 z^{q-1} + \cdots + \theta_q \quad (10.12)$$

lie inside the unit circle in the complex plane.

Remark 10.2. Similar to Remark 10.1 for the AR polynomial, if one writes MA polynomial as $\theta(z)$, that is

$$\mathscr{P}_\theta^\dagger(z) = 1 + \theta_1 z + \cdot + \theta_q z^q\,, \tag{10.13}$$

then the equivalent condition of invertibility in terms of (10.13) is that all roots lie outside of unit circle in the complex plane.

The ARMA Models

Stationary time series with complex autocorrelation behavior often are modeled by mixed autoregressive and moving average (ARMA) processes which can be more parsimonious than a pure AR or pure MA model.

A time series $\{Y_t\}_{t=\infty}^{\infty}$ is ARMA(p,q) is defined as

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q}\,,$$

where $c = \mu(1-\phi_1 - \cdots - \phi_p)$. More concisely, we can write it as

$$\phi(B)x_t = \theta(B)\varepsilon_t\,.$$

To ensure $\{Y_t\}$ to be stationary, causal and invertible, we will require that both $\mathscr{P}_\phi(z)$ and $\mathscr{P}_\theta(z)$, which are the AR and MA polynomials associated with $\phi(B)$ and $\theta(B)$, have all the roots lie inside the unit circle.

Parameter Redundancy There is a potential problem with the class of the ARMA models if $\mathscr{P}_\phi(z)$ and $\mathscr{P}_\theta(z)$ share any common factors which will cause parameter redundancy. This is a serious

problem, not only the model is over-parametrized containing a number of unnecessary parameters, estimating these parameters adds substantial errors to the estimated model and degrades any statistical inference or forecast based on it.

In addition to the requirements that the roots of both AR and MA polynomials $\mathscr{P}_\phi(z)$ in (10.8) and $\mathscr{P}_\theta(z)$ in (10.12) should lie inside the unit circle, an equally important requirement is no common factors in AR and MA polynomials $\mathscr{P}_\phi(z)$ and $\mathscr{P}_\theta(z)$. These three requirements are to assure our model is causal, invertible and without parameter redundancy.

Eg 10.5. *Parameter redundancy*. Consider the process

$$\begin{aligned} Y_t &= 0.4Y_{t-1} + 0.45Y_{t-2} + \varepsilon_t + \varepsilon_{t-1} + 0.25\varepsilon_{t-2}\,,\\ \text{or } (1-0.4B-0.45B^2)Y_t &= (1+B+0.25B^2)\varepsilon_t\,. \end{aligned}$$

in the operator form. It appears that Y_t is an ARMA(2,2) process. But, its associated AR and MA polynomials are

$$\begin{aligned} \mathscr{P}_\phi(z) &= z^2 - 0.4z - 0.45 = (z+0.5)(z-0.9)\\ \mathscr{P}_\theta(z) &= z^2 + z + 0.25 = (z+0.5)^2\,. \end{aligned}$$

The two polynomials share a common factor $(z+0.5)$ which can be canceled out. After cancellation, the polynomials become $\mathscr{P}_\phi(z) = (z-0.9)$ and $\mathscr{P}_\theta(z) = (z+0.5)$, so the model is an ARMA(1,1) model, $(1-0.9B)Y_t = (1+0.5B)\varepsilon_t$, or

$$Y_t = 0.9Y_{t-1} + 0.5\varepsilon_{t-1} + \varepsilon_t\,.$$

The model is causal because $\mathscr{P}_\phi(z) = (z - 0.9)$ has a root 0.9 which is inside the unit circle. the model is also invertible because the root of $\mathscr{P}_\theta(z) = (z + 0.5)$ is -0.5 , which also inside the unit circle.

The problem of parameter redundancy occurs very frequently in modeling financial data. The fact that most of them have ARCH effects in the series exacerbate the parameter redundancy problem. The difficult part of dealing the problem is that the estimated model will not appear exact same factors in the two polynomials. If we fail to check our estimated parameters carefully and blindly rely on a model selection criterion or auto selection tool, we will end up with a poor model. We shall discuss more about this problem later in this handout.

Partial autocorrelations The concept of partial autocorrelations resembles to the partial correlations of a regression model. For simplicity, we assume $\mu = 0$. The lag h partial autocorrelation ϕ_{hh} is the last coefficient of that minimizes

$$E\left(Y_t - \phi_{h1}Y_{t-1} - \cdots - \phi_{hh}Y_{t-h}\right)^2 .$$

For a given h, ϕ_{hh} can be obtained by solving the Yule Walker equations,

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_h \end{bmatrix} = \begin{bmatrix} \rho_0 & \rho_1 & \cdots & \rho_{h-1} \\ \rho_1 & \rho_0 & \cdots & \rho_{h-2} \\ \vdots & \vdots & & \vdots \\ \rho_{h-1} & \rho_{h-2} & \cdots & \rho_0 \end{bmatrix} \begin{bmatrix} \phi_{h1} \\ \phi_{h2} \\ \vdots \\ \phi_{hh} \end{bmatrix} \tag{10.14}$$

Or, in the matrix form, $\boldsymbol{\rho}_h = \boldsymbol{R}_h \boldsymbol{\phi}_h$. The solution is $\boldsymbol{\phi}_h = \boldsymbol{R}_h^{-1}\boldsymbol{\rho}_h$.

PACF of MA(1) The ACF $\rho_1 = \theta/\left(1+\theta^2\right)$ and $\rho_h = 0$, $h > 1$. Thus

$$\phi_{11} = \gamma_1/\gamma_0 = \rho_1 = \theta/\left(1+\theta^2\right) .$$

It can be shown that for the MA(1) process, the PACF at lag h is

$$\phi_{hh} = -\frac{(-\theta)^h}{1+\theta^2+\cdots+\theta^{2h}} .$$

The partial autocorrelation of an MA(1) decays as the lag h increases.

PACF of AR(1) The ACF $\rho_h = \phi^{|h|}$, thus

$$\phi_{11} = \phi \quad \text{and} \quad \phi_{22} = \frac{-\rho_1^2+\rho_2}{1-\rho_1^2} = \frac{-\phi^2+\phi^2}{1-\phi^2} = 0 .$$

It can be shown by equation (10.14) that $\phi_{hh} = 0$ for $h > 1$.

PACF of AR(p) For the causal AR(p) process has its pth and higher order PACF,

$$\phi_{pp} = \phi_p , \qquad \phi_{hh} = 0 , \quad h > p.$$

Furthermore, the pth equation of (10.14) $gives\left(\phi_{p1}, \phi_{p2}, \ldots, \phi_{pp}\right) = \left(\phi_1, \phi_2, \ldots, \phi_p\right)$. We see that the PACF of an AR(p) cuts off after lag p and the ACF of an MA(q) cuts off after lag q.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Table 10.1: The properties of ACF and PACF in the class of ARMA models.

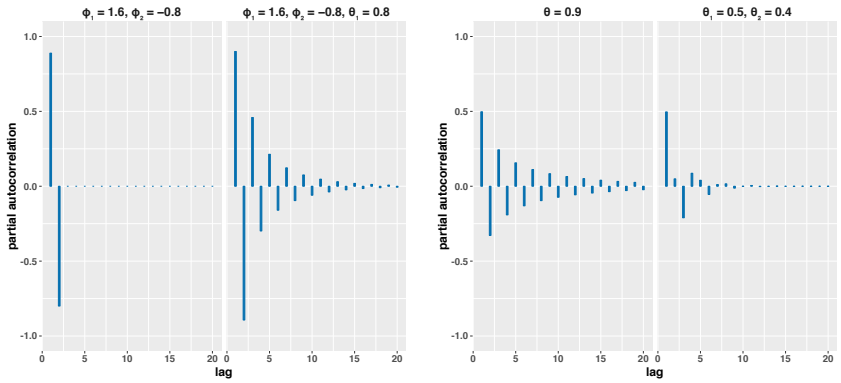


Figure 10.3: The left panel are partial autocorrelations of AR(2) and ARMA(2,1) models the same AR coefficiets. The right panel are partial autocorrelations of MA(1) and MA(2) models.

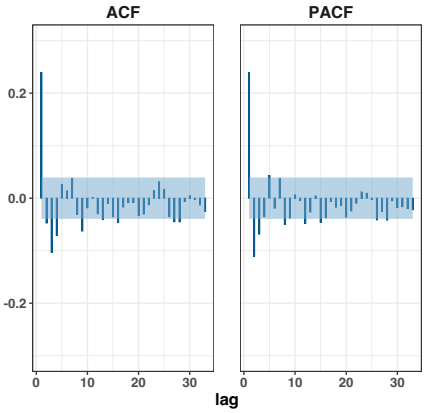
The first plot of the left panel in Figure 10.3 is the PACF of an AR(2) model, the 2nd plot is the PACF of an ARMA(2,1) with the same AR coefficients. We see the former cuts off after lag 2, the latter does not but decays rapidly. The right panel shows the PACF of MA(1) and MA(2) models, there is no cutoff but both decay fast.

Sample Partial Autocorrelations The sample partial autocorrelationscan be obtained by replacing the autocorrelations ρ_h in the Yule Walker equation system (10.14) with the sample autocorrelations $\hat{\rho}_h$.

Examining the sample ACF and sample PACF provides some guidance to identify the orders of an ARMA model. However, we should not expect the sample versions will behave like what are listed in Table 10.1 due to the sampling variability. In R, the pacf plot can be done by either `acf(x, type = "partial")` or simply `pacf(x)` .

Estimating ARMA Parameters The parameters of ARMA models can be estimated by maximum likelihood or conditional least-squares using the R function `arima()`. We will rely on model selection criterion such as the AIC or BIC to select a model. This would require a preliminary data analysis to list a set of candidate models.

Eg 10.6. The daily returns of Kodak from Eg. 10.3. Beyond the first order, the ACF has a more significant 3rd lag, while PACF has a more apparent 2nd lag. It is unlikely to select p and q from these plots which only suggest both ACF and PACF decay rapidly. But we can set the max values of p and q to be 2 or 3 to make the list of our candidate models.



For each model, we compute the AIC or BIC. Recall that recall that

$$\text{AIC} = -2\log \text{ML} + 2 \times \#\text{ of parameters}$$
$$\text{BIC} = -2\log \text{ML} + \log n \times \#\text{ of parameters}$$

The standard parameters are p ϕ's, q θ's and σ_ε^2, plus other parameters such as mean or trend. Thus, the total number of parameters ARMA(p,q) model with a mean is $p+q+2$, without the mean, $p+q+1$.

The log of maximum likelihood and AIC are included in the return values of R's `arima()`.

```
names(fit.ar3)

##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"
##  [6] "aic"       "arma"      "residuals" "call"      "series"
## [11] "code"      "n.cond"    "nobs"      "model"
```

Many statistical software packages have functions to automate the search for the ARMA model that optimizes AIC or BIC criteria such as the `auto.airma` function in R's `forecast` package.

```
library(forecast)
args(auto.arima)

## function (y, d = NA, D = NA, max.p = 5, max.q = 5, max.P = 2,
##     max.Q = 2, max.order = 5, max.d = 2, max.D = 1, start.p = 2,
##     start.q = 2, start.P = 1, start.Q = 1, stationary = FALSE,
##     seasonal = TRUE, ic = c("aicc", "aic", "bic"), stepwise = TRUE,
##     nmodels = 94, trace = FALSE, approximation = (length(x) >
##         150 | frequency(x) > 12), method = NULL, truncate = NULL,
##     xreg = NULL, test = c("kpss", "adf", "pp"), test.args = list(),
##     seasonal.test = c("seas", "ocsb", "hegy", "ch"), seasonal.test.args
##     = list(), allowdrift = TRUE, allowmean = TRUE, lambda = NULL,
##     biasadj = FALSE, parallel = FALSE, num.cores = 2, x = y, ...)
```

The `auto.airma()` function is built on the `arima()` function and can be used for a large class of extended ARMA models. An automatically selected model should not be accepted blindly, it helps to start with some model quickly. We should view the `auto.airma()` function as a tool to save time to compute and fit models one by one. The proper way of building a model should always start with exploring data to identify several candidate models.

The argument `ic` stands for information criteria, is for setting which criterion to use. The default is AICc which is the corrected AIC

to "correct" the overfitting problem of the AIC in finite sample. It makes no difference form the AIC for the sample size we use.

Eg 10.7. *Pitfalls of automatic selection*. Continue the Kodak daily return data. The default of `auto.arima()` sets `max.p = 5` and `max.q = 5`. The function selects a model that has lowest AICc value among ARMA(p,q) models with all possible combinations of $0 \leq p,q \leq 5$ excluding those models with roots being close to the unit circle. Suppose that we simply use it for our data without setting appropriate upper bounds for p and q, we get the following results.

```
autofit = auto.arima(yt); autofit

## Series: yt
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##          ar1      ma1      ma2
##       0.9222  -0.6763  -0.2761
## s.e.  0.0217   0.0271   0.0190
##
## sigma^2 = 33.97:  log likelihood = -7873.08
## AIC=15754.16   AICc=15754.17   BIC=15777.41
```

The model selected by `auto.arima()` is an ARMA(1,2) model. The default setting estimates mean, "zero mean" means insignificance. All three coefficients are significant. We can calculate the roots o AR and MA polynomials using the `R` function `polyroot()`. The function takes a vector $\boldsymbol{a} = (a_0, \ldots, a_k)^T$, solves zero roots for a polynomial $a_0 + a_1 z + \cdots + a_k z^k$.

```
coef(autofit)
```

```
##         ar1         ma1         ma2
##  0.9222441 -0.6763166 -0.2760752
```

```
ar = coef(autofit)[1]; ma = coef(autofit)[2:3]
ar.roots = polyroot(rev(c(1,-ar)));
ma.roots = polyroot(rev(c(1,ma)));
cat("AR Roots:", round(ar.roots,4)); cat("MA Roots:", round(ma.roots,4))
```

```
## AR Roots: 0.9222+0i
## MA Roots: -0.2867+0i 0.963+0i
```

One of the MA root 0.963 is very close to the AR root 0.922. We can also see how close they in the left root plot of Figure 10.4. This is a clear sign of parameter redundancy, the ARMA(1,2) model should be excluded from consideration.

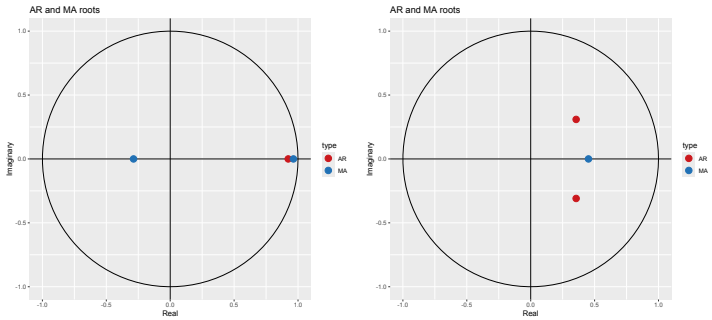


Figure 10.4: Fitting ARMA models to daily return of Kodak. Left: The AR and MA roots of ARMA(1,2) model. Right: The AR and MA roots of ARMA(2,1) model.

The root plot of ARMA(1,2) model from `auto.arima()` suggests that we should not consider models with $p+q>3$. You can choose to fit models with $p+q=3$ except (1,2). Introduce `forecast`' function `Arima()` which is essentially the same as the base R function

`arima()` but with return values of all three model selection criteria AIC, AICc and BIC instead of only the AIC.

```
ps = 0:3; qs = 0:3
AIC =  BIC = matrix(nrow = length(ps), ncol = length(qs))
rownames(AIC) = rownames(BIC) = paste0("p = ", ps)
colnames(AIC) = colnames(BIC) = paste0("q = ", qs)
for(i in ps){
  for(j in qs) {
        if((i + j)<= 3){
            arma = Arima(yt, order = c(i,0,j))
            AIC[i+1,j+1] = arma$aic; BIC[i+1,j+1] = arma$bic
        }
  }
}
AIC;
```

```
##          q = 0    q = 1    q = 2    q = 3
## p = 0 15941.10 15778.02 15779.68 15770.94
## p = 1 15797.63 15779.81 15755.48       NA
## p = 2 15769.08 15759.67       NA       NA
## p = 3 15759.42       NA       NA       NA
```

```
BIC
```

```
##          q = 0    q = 1    q = 2    q = 3
## p = 0 15952.73 15795.47 15802.94 15800.01
## p = 1 15815.07 15803.06 15784.55       NA
## p = 2 15792.34 15788.74       NA       NA
## p = 3 15788.49       NA       NA       NA
```

Excluding ARMA(1,2), both AIC and BIC select the AR(3) model followed by ARMA(2,1) and AR(2). The right plot in Figure 10.4 shows the root plot of ARMA(2,1).

```
ar3 = Arima(yt, order = c(3,0,0)); ar3
```

```
## Series: yt
## ARIMA(3,0,0) with non-zero mean
##
```

```
## Coefficients:
##          ar1      ar2      ar3     mean
##       0.2578  -0.0925  -0.0685  -0.0607
## s.e.  0.0200   0.0206   0.0201   0.1297
##
## sigma^2 = 34.03:  log likelihood = -7874.71
## AIC=15759.42   AICc=15759.45   BIC=15788.49
```

```
names(ar3)
```

```
##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
##  [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
## [13] "nobs"      "model"     "aicc"      "bic"       "x"         "fitted"
```

All three AR coefficients are significant, though the mean is not significant, in convention, we would model with a mean nevertheless. Both `arima()` and `Arima()` has an argument `include.mean` that we can preset to include or exclude the mean in model selection. While `auto.arima()` views a model with and without the mean as two candidate models.

Residual Analysis

In practice, unlike our example, after thoroughly examining the selected models and excluding those problematic ones, the AIC and BIC often end up with different models. Diagnostic analysis with model residuals can be helpful in deciding a final model. We may also want to consider those models with orders in between the AIC and BIC selected models. If the analysis results are similar, we should choose the model with the fewest parameters.

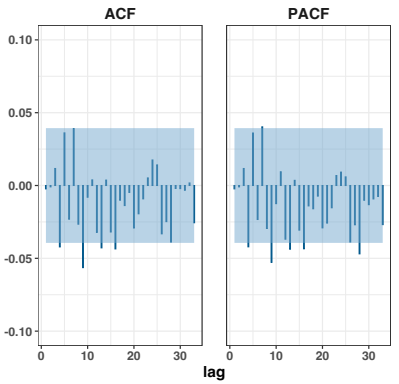
We can plot the ACF/PACF of residuals and test for remaining correlation in residuals with the Ljung-Box test. When the Ljung-Box

test is applied to residuals, a correction for the degrees of freedom is needed. If the residuals are from fitting an ARMA(p,q) model, the $Q(K)$ statistic has degrees of freedom df $= K-p-q$ under the null hypothesis that the residuals are not correlated. In using R function `Box.test()`, the correction is done by setting `fitdf = p+q`. The lag we suggested earlier $\log n$ should also have the same adjustment, K should be such that $K-p-q = \text{DF} = \log n$.

The ACF and PACF of residuals show little significance. Yet the Box-Ljung test for residuals from fitting AR(3) at DF $= \log n \approx 8$ is significant at 1% implying correlation in residuals. The tests for lags 4 - 20, corresponding to DF 1 - 16, are significant at 1% for DF > 6 (lags > 10).

```
res = ar3$res;
pq = sum(ar3$arma[1:2])
Box.test(res, round(log(n)) + pq,
         type ="L", fitdf = pq)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 22.997, df = 8,
## p-value = 0.003368
```



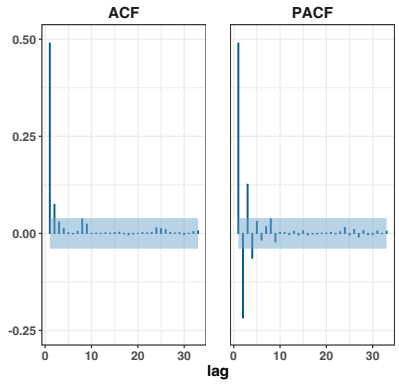

```
df = (1:16); pv = c()
for(i in df){
  pv[i] = Box.test(res,i + pq, "L", pq)$p.val
}
names(pv) = df; round(pv,5)
```

```
##       1       2       3       4       5       6       7       8       9
## 0.02898 0.01820 0.02506 0.01064 0.01085 0.00087 0.00173 0.00337 0.00240
##      10       11      12      13      14      15      16
## 0.00081  0.00148 0.00107 0.00035 0.00058 0.00085 0.00140
```

For asset return series, we also expect ARCH/GARCH effect in the data. This will be discussed in the next handout. When the shocks ε_t are ARCH/GARCH white noise. the estimating error $1/\sqrt{n}$ of $\hat{\rho}_h$ given in (10.1), for testing testing $\rho_h = 0$ is underestimated. Consequently, we will see false significance in the ACF plots and Ljung Box tests which are sum of squared sample ACFs.

One quick way to check the existence of ARCH effect is plotting the ACF of the squared return data. If the plot exhibits clear significance in several ACFs, it is an indication of ARCH effect in the data.

The sample ACF plot of squared residuals of fitting AR(3) to the Kodak daily returns shows obvious significance at first lag and the PACF plot up to lag 4. The evidence of dependence among squared residuals, and thus ARCH effect in the data is apparent.



A stock return series can be modeled more adequately with an ARMA (p,q) + ARCH/GARCH model. For the class of ARMA(p,q) which are conditional mean models only, the ACF and PACF of AR(3) residuals show only little significance, we conclude that the model for the Kodak daily returns is the AR(3) model, with AR coefficient estimates, $\hat{\phi}_1 = 0.2678$, $\hat{\phi}_2 = -0.0925$, $\hat{\phi}_3 = -0.0685$, and the variance of shock $\hat{\sigma}^2_\varepsilon = 34.03$.

Forecasting for ARMA Models

Once a model has been selected and fitted to the data, forecasting future values of the time series can be carried out. That is, predict future values, Y_{n+k}, $k = 1, 2, \ldots$, based on $Y_1, \ldots, Y_n$. Let $\mathscr{F}_n$ denote the information generated by the sequence $\{Y_t\}, 1 \le t \le n$. The one-step prediction for Y_{n+1} is the expectation conditional on $\mathscr{F}_n$, that is, $E[Y_{n+1}|\mathscr{F}_n]$.

AR(1) Models Consider forecast of an an AR(1) model,

$$Y_{n+1} - \mu = \phi(Y_n - \mu) + \varepsilon_{n+1}.$$

The one-step predictor of Y_{n+1} at time n is

$$Y^n_{n+1} = E[Y_{n+1}|\mathscr{F}_n] = \mu + E[\phi(Y_n - \mu) + \varepsilon_{n+1}|\mathscr{F}_n] = \mu + \phi(Y_n - \mu),$$

because $E(\varepsilon_{n+k}|\mathscr{F}_n) = 0$, $k \ge 1$. The 2-step predictor is

$$Y^n_{n+2} = E[Y_{n+2}|\mathscr{F}_n] = \mu + E[\phi(Y_{n+1} - \mu)|\mathscr{F}_n] = \mu + \phi^2(Y_n - \mu).$$

Continuing in this way, the k-step predictor is

$$Y^n_{n+k} = E[Y_{n+k}|\mathscr{F}_n] = \mu + \phi^k(Y_n - \mu).$$

If $|\phi| < 1$, as is true for a stationary series, the forecasts will converge exponentially fast to μ, the mean of Y_t, as k increase.

The k-step-ahead forecasts based on the estimated model are simply replacing μ and ϕ by their estimated counterparts, $\hat{\mu}$ and $\hat{\phi}$,

$$\hat{Y}^n_{n+k} = \hat{\mu} + \hat{\phi}^k(Y_n - \hat{\mu}).$$

Forecast errors of AR(1) The mean square prediction error of 1-step predictor is

$$E[Y_{n+1}-Y_{n+1}^n]^2 = E\big[\{\mu+\phi(Y_n-\mu)+\varepsilon_{n+1}\}-\{\mu+\phi(Y_n-\mu)\}\big]^2 = E[\varepsilon^2]=\sigma_\varepsilon^2,$$

The $(1-\alpha)100\%$ 1-step-ahead prediction interval is

$$\mu+\phi(Y_n-\mu)\pm z_{\alpha/2}\sigma_\varepsilon\,, \tag{10.15}$$

if $\{\varepsilon_t\}$ are Gaussian. Adjust $z_{\alpha/2}$ for a different distribution of ε_t.

The error in the two-step ahead prediction is

$$Y_{n+2}-Y_{n+2}^n=\phi(Y_{n+1}-Y_{n+1}^n)+\varepsilon_{n+1}=\phi\varepsilon_{n+1}+\varepsilon_{n+2}\,.$$

The corresponding mean square prediction error is $(1+\phi^2)\sigma_\varepsilon^2$ and prediction interval is $Y_{n+2}^n\pm z_{\alpha/2}(1+\phi^2)^{1/2}\sigma_\varepsilon$ respectively. The k-step ahead error and its variance are

$$Y_{n+k}-Y_{n+k}^n=\sum_{i=0}^{k-1}\phi^i\varepsilon_{n+k-i}\quad\text{and}$$

$$(\theta^{2(k-1)}+\cdots+\phi^2+1)\sigma_\varepsilon^2\longrightarrow\frac{\sigma_\varepsilon^2}{1-\phi^2}=\gamma_0\quad\text{as}\quad k\to\infty\,.$$

The mean square prediction error converges to the variance of $\{Y_t\}$, γ_0 exponentially fast as k in creases. The $(1-\alpha)100\%$ k-step-ahead prediction interval converges rapidly to

$$\mu\pm z_{\alpha/2}\sqrt{\gamma_0}\,. \tag{10.16}$$

Based on an estimated model, $\hat\phi$, $\hat\mu$, $\hat\sigma_\varepsilon$ and $\hat\gamma_0$ will be used instead.

AR(p) models By the same derivation, the k-step-predictor of an AR(p) model is

$$Y_{n+k}^n=\mu+\sum_{i=1}^{p}\phi_i\{(Y_{n+k-i}\mathbf{1}_{\{n+k-i\le n\}}+Y_{n+k-i}^n\mathbf{1}_{\{n+k-i>n\}})-\mu)\}\,,$$

where $\mathbf{1}_{\{\cdot\}}$ is indicator function. Assuming Gaussianarity, the 1-step-prediction interval is

$$Y_{n+1}^n\pm z_{\alpha/2}\sigma_\varepsilon=\{\phi_1(Y_n-\mu)+\cdots+\phi_p(Y_{n+1-p}-\mu)\}\pm z_{\alpha/2}\sigma_\varepsilon\,.$$

As $k\to\infty$, $Y_{n+k}^n\to\mu$ and the variance of the prediction error converge to γ_0. The k-step-prediction interval converge to (10.16).

MA(1) Models If Y_t is an MA(1) model, $Y_t=\mu+\theta\varepsilon_{t-1}+\varepsilon_t$, the one-step predictor is simply

$$Y_{n+1}^n=E[Y_{n+1}|\mathscr{F}_n]=\mu+E[\varepsilon_{n+1}+\theta\varepsilon_n|\mathscr{F}_n]=\mu+\theta\varepsilon_n\,.$$

For the k-step ahead predictor, $k\ge 2$, the conditional expectation

$$Y_{n+k}^n=E[Y_{n+k}|\mathscr{F}_n]=\mu+E[\varepsilon_{n+k}+\theta\varepsilon_{n+k-1}|\mathscr{F}_n]=\mu\,,\qquad k\ge 2.$$

In practice, $\hat\mu$, $\hat\theta$ and residual $\hat\varepsilon_n$ will be used instead.

MA(q) Models The k-step-ahead forecast of an MA(q), $Y_t=\mu+\varepsilon_t+\theta_1\varepsilon_{t-1}+\cdots+\theta_q\varepsilon_{t-q}$, is based on the conditional expectation,

$$Y_{n+k}^n=\mu+\sum_{i=1}^{q}\theta_iE[\varepsilon_{n+k-i}|\mathscr{F}_n]=\begin{cases}\mu+\sum_{i=k}^{q}\theta_k\varepsilon_{n+k-i} & ,\ k\le q\\ \mu & ,\ k>q\end{cases}$$

Thus, the mean square prediction error is

$$E[(Y_{n+k} - Y_{n+k}^n)^2] = E\Big[\Big(\varepsilon_{n+k} + \sum_{i=1}^{k-1} \theta_i \varepsilon_{n+k-i}\Big)^2\Big]$$
$$= \begin{cases} (1 + \sum_{i=1}^{k-1} \theta_i^2)\sigma_\varepsilon^2 & , k \le q \\ \gamma_0 & , k > q \end{cases}$$

The $(1-\alpha)100\%$ prediction interval for $k > q$ is $\mu \pm z_{\alpha/2}\sqrt{\gamma_0}$, the same as (10.16). As before, our forecasts will be based on the estimated model, $\hat{\mu}$, $\hat{\theta}_i, i = 1, \ldots, q$ and residuals $\hat{\varepsilon}_{n-i}$ will be used instead.

ARMA(p,q) Models The 1-step-ahead forecast of an ARMA(p,q) model is

$$Y_{n+1}^n = \mu + \sum_{i=1}^{p} \phi_i (Y_{n+1-i} - \hat{\mu}) + \sum_{i=1}^{q} \theta_i \varepsilon_{n+1-i}$$

and the 1-step-ahead prediction error is ε_{n+1} with has mean square σ_ε^2. The k-step predictor is a combination of what have been derived for AR(p) and MA(q)

$$Y_{n+k}^n = \mu + \sum_{i=1}^{p} \phi_i \{(Y_{n+k-i} \mathbf{1}_{\{n+k-i \le n\}} + Y_{n+k-i}^n \mathbf{1}_{\{n+k-i > n\}}) - \mu)\}$$
$$+ \sum_{i=k}^{q} \theta_i \varepsilon_{n+k-i} \mathbf{1}_{\{k \le q\}}$$

The prediction error can be calculated using the MA(∞) representation of the model. As k increases the k-step-ahead forecast, $Y_{n+k}^n \to \mu$ and the mean square prediction error converges to γ_0.

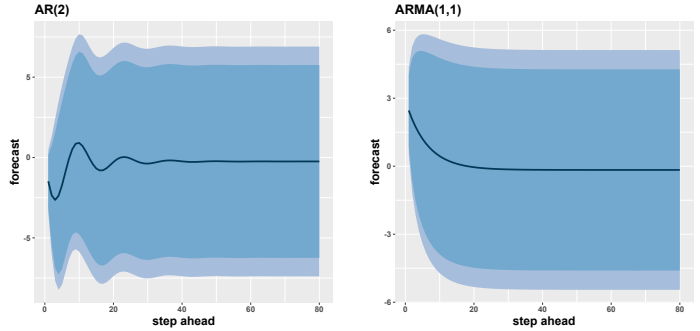


Figure 10.5: Forecasts, 90% and 95 % forecast intervals of simulated AR(2) with $\phi_1 = 1.6, \phi_2 = 0.8$ and ARMA(1,1) with $\phi = 0.85$ and $\theta = 0.5$.

Figure 10.5 shows the k-step ahead forecasts of a simulated AR(2) and ARMA(1,1) along with the corresponding 90% and 95% prediction intervals. The forecasts become constant and prediction intervals become fixed as k increases.

For any causal and invertible ARMA model, the k-step-ahead predictor converges to the mean of the process μ and the mean square prediction error converges to the variance of the process γ_0.

Forecast ARMA in R The R function `predict()` gives the forecast and forecast standard error from an ARMA fit. The number of steps should be given in the `n.ahead` argument.

```
pred  = predict(ar3, n.ahead = 10); pred

## $pred
## Time Series:
## Start = 2476
## End = 2485
## Frequency = 1
##  [1] -0.07110588  0.27936890  0.13891150 -0.04003582 -0.09718181 -0.08572609
##  [7] -0.06522136 -0.05707987 -0.05766393 -0.05997311
```

```
##
## $se
## Time Series:
## Start = 2476
## End = 2485
## Frequency = 1
##  [1] 5.833120 6.023776 6.025699 6.053370 6.058046 6.058047 6.058367 6.058452
##  [9] 6.058453 6.058456
```

We can also use `forecast`'s R function `forecast()` for prediction. The `h` argument is the same as n.ahead of `predict()`. This function also gives prediction intervals assuming Gaussianity on the process. The prediction levels can be set with the argument `level`, the default prediction levels are 80% and 95%.

```
fore = forecast(ar3, h = 10,level = c(90,95)); fore
```

```
##      Point Forecast      Lo 90     Hi 90     Lo 95    Hi 95
## 2476    -0.07110588  -9.665735  9.523523 -11.50381 11.36160
## 2477     0.27936890  -9.628860 10.187598 -11.52701 12.08575
## 2478     0.13891150  -9.772482 10.050305 -11.67124 11.94906
## 2479    -0.04003582  -9.996944  9.916872 -11.90442 11.82435
## 2480    -0.09718181 -10.061781  9.867417 -11.97073 11.77637
## 2481    -0.08572609 -10.050326  9.878874 -11.95928 11.78783
## 2482    -0.06522136 -10.030348  9.899906 -11.93940 11.80896
## 2483    -0.05707987 -10.022347  9.908187 -11.93143 11.81727
## 2484    -0.05766393 -10.022932  9.907604 -11.93201 11.81669
## 2485    -0.05997311 -10.025247  9.905301 -11.93433 11.81438
```

```
names(fore)
```

```
##  [1] "method"    "model"     "level"     "mean"      "lower"     "upper"
##  [7] "x"         "series"    "fitted"    "residuals"
```

The point forecasts are given in `$mean`. The print output of `arima()` or `Arima()` is in `$model`.

Apply `plot()` or `autoplot()` to a forecast output object like `fore` will plot the point forecasts and prediction intervals along with the original time series. In the `forecast` package, `autoplot()` plots the ggplot version of a plot.

Appendix: Daily Returns of Kellanova (formally Kellogg)

There are three series in Eg. 10.3, we have concluded that the Bristol-Myers Squibb daily returns are a white noise process and the Kodak ones can be modeled with an ARMA model. This appendix will focus on the Kellogg daily returns. The Ljung-Box tests show significance at all lags tested except the 2nd lag. The ACF plot show weak significance at a few lags. This is a sign of ARCH/GARCH errors which are a non independent white noise process.

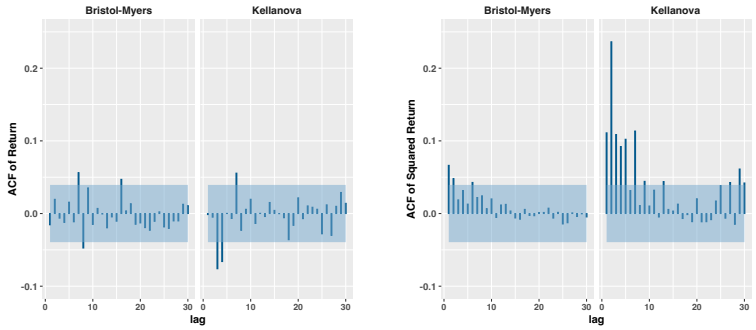


Figure 10.6: Left: The ACF of daily returns of Bristol-Myers and Kellogg, the same ACF plots in Figure 10.1. Right: The ACF of squared daily returns of Bristol-Myers and Kellogg

When the white noise are not *i.i.d.* the 95% CI band of $1.96/\sqrt{n}$ is underestimated. This explains the significance seen in the ACF of Kellogg daily returns at the left plot of Figure 10.6. The right plot is the ACF of squared daily return of Bristol-Myers and Kellogg. Kellogg's squared daily returns show strong evidence of serial correlations. The difference between Kellogg and Bristol-Myers in both sets of ACF plots represents the difference between independent and

non-independent white noise processes.

Fitting Kellogg's daily returns to an ARMA model, will not get any model with significant AR or MA coefficients as shown here. Furthermore, the white noise model has the lowest AIC and BIC scores.

```
## **** Fitting AR(1) to daily returns of Kellogg ****
Arima(Y3[,"K"], order = c(1,0,0))

## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1   mean
##       -0.001  0.011
## s.e.   0.020  0.028
##
## sigma^2 = 1.89:  log likelihood = -4299.49
## AIC=8604.98   AICc=8604.99   BIC=8622.42

## **** Fitting MA(1) to daily returns of Kellogg ****
Arima(Y3[,"K"], order = c(0,0,1))

## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1   mean
##       -0.001  0.011
## s.e.   0.020  0.028
##
## sigma^2 = 1.89:  log likelihood = -4299.49
## AIC=8604.98   AICc=8604.99   BIC=8622.42

## **** Fitting white noise to daily returns of Kellogg ****
Arima(Y3[,"K"], order = c(0,0,0))

## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.011
## s.e.  0.028
##
## sigma^2 = 1.89:  log likelihood = -4299.49
## AIC=8602.98   AICc=8602.99   BIC=8614.61
```