Lecture 14
ARIMA – Identification, Estimation & Seasonalities

 1

ARMA Process

• We defined the ARMA(p, q) model:

$$\phi(L)(y_t - \mu) = \theta(L)\varepsilon_t$$

Let $(y_t - \mu) = x_t$

Then, $\phi(L)x_t = \theta(L)\varepsilon_t$

$\Rightarrow$ x_t is a *demeaned* ARMA process.

• In this lecture, we will study:
- Identification of p, q.
- Estimation of ARMA(p, q)
- Non-stationarity of x_t.
- Differentiation issues – ARIMA(p, d, q)
- Seasonal behavior – SARIMA(p, d, q)$_S$

Autocovariance Function

- We define the autocovariance function, $\gamma(t-j)$ as:

$$\gamma(t-j=k) = E[\, y_t, y_{t-j}]$$

- For an AR(p) process, WLOG with μ =0 (or demeaned y_t), we get:

$$\gamma(k) = E[(\phi_1\, y_{t-1}y_{t-k} + \phi_2 y_{t-2}y_{t-k} + \ldots + \phi_p y_{t-p}y_{t-k} + \varepsilon_t y_{t-k})]$$
$$= \phi_1\,\gamma(k-1) + \phi_2\gamma(k-2) + \ldots + \phi_p\gamma(k-p)$$

Notation: $\gamma(k)$ is commonly used. Sometimes, $\gamma(k)$ is referred as "covariance at lag k.

- The $\gamma(k)$ determine a system of equations:

$$\gamma(0) = E[y_t, y_t] = \phi_1\,\gamma(1) + \phi_2\,\gamma(2) + \phi_3\,\gamma(3) + \ldots + \phi_p\,\gamma(p) + \sigma^2$$
$$\gamma(1) = E[y_t, y_{t-1}] = \phi_1\,\gamma(0) + \phi_2\,\gamma(1) + \phi_3\,\gamma(2) + \ldots + \phi_p\,\gamma(p-1)$$
$$\gamma(2) = E[y_t, y_{t-2}] = \phi_1\,\gamma(1) + \phi_2\,\gamma(0) + \phi_3\,\gamma(1) + \ldots + \phi_p\,\gamma(p-2)$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

ACF: Estimation (System of Equations)

- The pxp system of equations:

$$\gamma(1) = E[y_t, y_{t-1}] = \phi_1\gamma(0) + \phi_2\gamma(1) + \phi_3\gamma(2) + \ldots + \phi_p\gamma(p-1)$$
$$\gamma(2) = E[y_t, y_{t-2}] = \phi_1\gamma(1) + \phi_2\gamma(0) + \phi_3\gamma(1) + \ldots + \phi_p\gamma(p-2)$$
$$\gamma(3) = E[y_t, y_{t-3}] = \phi_1\gamma(2) + \phi_2\gamma(1) + \phi_3\gamma(0) + \ldots + \phi_p\gamma(p-3)$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

Using linear algebra, we write the system as: $\boldsymbol{\gamma} = \boldsymbol{\Gamma}\,\boldsymbol{\phi}$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(0) \end{bmatrix} \quad \text{a } p\text{x}p \text{ matrix}$$

$\boldsymbol{\phi}$ is the px1 vector of AR(p) coefficients
$\boldsymbol{\gamma}$ is the px1 vector of $\gamma(k)$ autocovariances.

ACF: Estimation – Yule-Walker

• Now, we define the autocorrelation function (**ACF**):

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{\text{covariance at lag } k}{\text{variance}}$$

The ACF lies between -1 and +1, with $\rho(0) = 1$.

• Dividing the autocovariance system by γ(0), we get:

$$\begin{bmatrix} \rho(0) & \rho(1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{bmatrix}$$

Or using linear algebra: $\mathbf{P}\,\boldsymbol{\phi} = \boldsymbol{\rho}$

• These are **Yule-Walker** equations, which can be solved numerically.

ACF: Estimation – Yule-Walker

• The Yule-Walker equations can be solved numerically. MM can be used (replace population moments with sample moments).

• Properties for a stationary time series

1. $\gamma(0) \geq 0$ (from definition of variance)
2. $\gamma(k) \leq \gamma(0)$ (from Cauchy-Schwarz)
3. $\gamma(k) = \gamma(-k)$ (from stationarity)
4. $\mathbf{\Gamma}$, the auto-correlation matrix, is psd ($a' \mathbf{\Gamma}\, a \geq 0$)

Moreover, any function γ : Z → R that satisfies (3) and (4) is the autocovariance of some stationary time series.

ACF: ARMA(1, 1) – Recursive Formula

- For an ARMA(1, 1) we have:.

$$y_t = \mu + \phi_1\, y_{t-1} + \theta_1\, \varepsilon_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

- **Moments:** (μ = 0)

$E[y_t] = \mu\ /\ (1 - \phi_1) = 0$ (assuming $\phi_1 \neq 1$)

$Var[y_t] = \sigma^2\,(1 + \theta_1^2\,)\ /\ (1 - \phi_1^2)$ (assuming $|\phi_1| < 1$)

- **Autocovariance function** (μ = 0)

$$\begin{aligned}\gamma(k) &= Cov[y_t,\ y_{t-k}]\\ &= E[\{\phi_1\, y_{t-1} + \theta_1\ \varepsilon_{t-1} + \varepsilon_t\}\, y_{t-k}]\\ &= \phi_1\, E[y_{t-1}\, y_{t-k}] + \theta_1\, E[\varepsilon_{t-1}\, y_{t-k}] + E[\varepsilon_t\, y_{t-k}]\\ &= \phi_1\, \gamma(k-1)\ + \theta_1\, E[\varepsilon_{t-1}\, y_{t-k}] + E[\varepsilon_t\, y_{t-k}]\end{aligned}$$

- Again, we have a **recursive formula**.

$$\gamma(k) = \phi_1\, \gamma(k-1)\ + \theta_1\, E[\varepsilon_{t-1}\, y_{t-k}] + E[\varepsilon_t\, y_{t-k}]$$

ACF: ARMA(1, 1) – Recursive Formula

- We have a recursive formula:

$$\gamma(k) = \phi_1\, \gamma(k-1) + E[\varepsilon_t\, \boldsymbol{y_{t-k}}] + \theta_1\, E[\varepsilon_{t-1}\, y_{t-k}]$$

It can be shown, after a lot of algebra:

For k = 0,

$$\gamma(0) = \phi_1\, \gamma(1) + \sigma^2 + \theta_1\, (\phi_1\, \sigma^2 + \theta_1 \sigma^2)$$

For k = 1,

$$\gamma(1) = \phi_1\, \gamma(0) + \theta_1\, \gamma(1)$$

For k = 2,

$$\gamma(2) = \phi_1\, \gamma(1)$$

For k,

$$\gamma(k) = \phi_1^{k-1}\, \gamma(1), \quad k > 1$$

$\Rightarrow$ If $|\phi_1|$<1, exponential decay.

ACF: ARMA(1, 1) – Stationarity

- Two equations for $\gamma(0)$ and $\gamma(1)$:

$\gamma(0) = \phi_1\,\gamma(1) + \sigma^2 + \theta_1\,(\phi_1\,\sigma^2 + \theta_1\,\sigma^2)$

$\gamma(1) = \phi_1\,\gamma(0) + \theta_1\,\gamma(1)$

Solving for $\gamma(0)$ & $\gamma(1)$:

$$\gamma(0) = \sigma^2\,\frac{1 + \theta_1^{\,2} + 2\,\phi_1\theta_1}{1 - \phi_1^{\,2}}$$

$$\gamma(1) = \sigma^2\,\frac{(1 + \phi_1\theta_1) * (\phi_1 + \theta_1)}{1 - \phi_1^{\,2}}$$

$$\vdots$$

$$\gamma(k) = \phi_1^{\,k-1}\,\gamma(1), \quad k > 1 \qquad \Rightarrow \text{If } |\phi_1| < 1\text{, exponential decay.}$$

Note: If stationary, ARMA(1,1) & AR(1) show exponential decay. Difficult to distinguish one from the other through autocovariances.

ACF: Estimation & Correlogram

- **Estimation**:

Easy: Use sample moments to estimate $\gamma(k)$ and plug in formula:

$$r_k = \hat{\rho}_k = \frac{\sum(Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum(Y_t - \bar{Y})^2}$$

We plug $\hat{\rho}_k = r_k$ in the Yule-Walker equations and solve for $\boldsymbol{\phi}$:

$$\boldsymbol{R}\,\boldsymbol{\phi} = \boldsymbol{r} \qquad \Rightarrow \widehat{\boldsymbol{\phi}} = \boldsymbol{R}^{-1}\,\boldsymbol{r}$$

where $\boldsymbol{R}$ is the estimated correlation matrix $\mathbf{P}$.

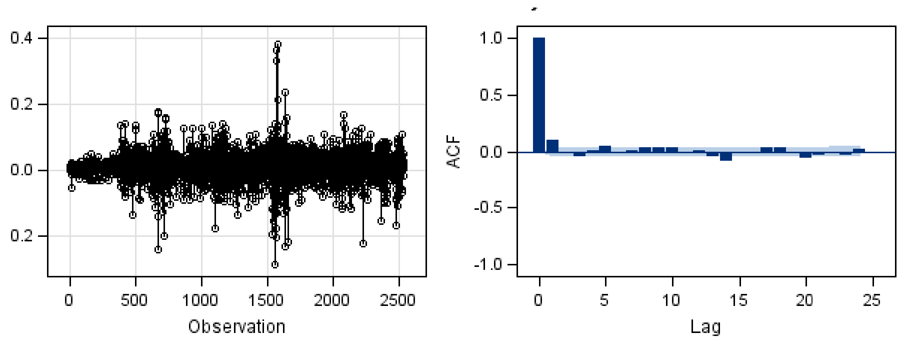
- The sample **correlogram** is the plot of the ACF against k. As the ACF lies between -1 and +1, the correlogram also lies between these values.

ACF: Estimation & Correlogram

• The sample correlogram is the plot of the ACF against *k*.

As the ACF lies between -1 and +1, the correlogram also lies between these values.

Example: Correlogram for US Monthly Returns (1800 – 2013)



ACF: Distribution

• **Distribution**:
For a linear, stationary process, $y_t = \mu + \sum_{j=0}^{\infty} \psi_j \, \varepsilon_{t-j}$,
with $E[\varepsilon_t^{\,4}] < \infty$, the distribution of the sample ACF, $r_k = \hat{\rho}_k$ is approximately normal with:

$$\mathbf{r} \xrightarrow{d} N(\boldsymbol{\rho}, \mathbf{V}/T), \qquad \mathbf{V} \text{ is the covariance matrix.}$$

Under H_0 (no autocorrelations) $\rho_k = 0$ for all $k > 1$.

$$\mathbf{r} \xrightarrow{d} N(\mathbf{0}, \mathbf{I}/T) \qquad \Rightarrow \text{Var}[r_k] = 1/T.$$

• Under H_0, the SE = $1/\sqrt{T}$ $\qquad$ **⇒ 95% C.I.: 0 ± 1.96 * $1/\sqrt{T}$**

Then, for an uncorrelated, WN sequence, approximately 95% of the sample ACFs should be within the above C.I. limits.

Note: The SE = $1/\sqrt{T}$ are sometimes referred as *Bartlett's SE*.

ACF – Identification

• The ACF can be used as a tool to select an ARMA(p, q) model. In general, it is used to select the lag q in an MA model.

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	0 after lag q	Tails off

Note: Ideally, "Tails off" is exponential decay. In practice, we may see decay with a lot of "noise" and a lot of non-zero values.

• In the next slides, we simulate ARMA models. This is an "ideal" situation, we know the model that generated the data. Then, we look at the ACF to see if it is easy to guess the model and order of the model.

ACF: MA(q)

Example: Sample ACF for an MA(q) process:

$$y_t = \mu + \varepsilon_t + \theta_1\ \varepsilon_{t-1} + \theta_2\ \varepsilon_{t-2} + \ldots + \theta_q\ \varepsilon_{t-q}$$

$$\rho(k) = \frac{\sum_{j=k}^{q} \theta_j \theta_{j-k}}{(1 + \theta_1{}^2 + \theta_2{}^2 + \ldots + \theta_q{}^2)} \qquad k \le q$$

$$= 0 \qquad \text{otherwise.}$$

For different k's:

$$\rho(0) = 1$$

$$\rho(1) = \frac{\theta_1 + \theta_2\theta_1 + \theta_3\theta_2}{(1 + \theta_1{}^2 + \theta_2{}^2 + \theta_3{}^2)}$$

$$\rho(2) = \frac{\theta_2 + \theta_3\theta_1}{(1 + \theta_1{}^2 + \theta_2{}^2 + \theta_3{}^2)}$$

$$\rho(3) = \frac{\theta_3}{(1 + \theta_1{}^2 + \theta_2{}^2 + \theta_3{}^2)}$$

$$\rho(k) = 0 \qquad \text{for } |k| > 3.$$

ACF: MA(q = 3)

Example (continuation):

$$y_t = \mu + \varepsilon_t + \theta_1\ \varepsilon_{t-1} + \theta_2\ \varepsilon_{t-2} + \theta_3\ \varepsilon_{t-3}$$

Suppose θ_1 = 0.5; θ_2 = 0.4; θ_3 = 0.2. Then,

$$\rho(0) = 1$$

$$\rho(1) = \frac{\theta_1 + \theta_2\theta_1 + \theta_3\theta_2}{(1 + {\theta_1}^2 + {\theta_2}^2 + {\theta_3}^2)} = \frac{0.5{+}0.4{*}0.5{+}0.1{*}0.4}{1 + 0.5^2 + 0.4^2 + 0.1^2} = \mathbf{0.5211}$$

$$\rho(2) = \frac{\theta_2 + \theta_3\theta_1}{(1 + {\theta_1}^2 + {\theta_2}^2 + {\theta_3}^2)} = \frac{0.4 + 0.1{*}0.5}{1 + 0.5^2 + 0.4^2 + 0.1^2} = \mathbf{0.3169}$$

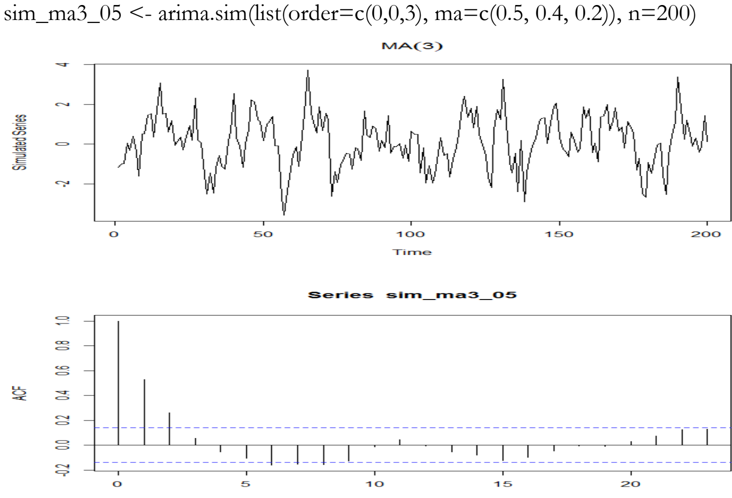
$$\rho(3) = \frac{\theta_3}{(1 + {\theta_1}^2 + {\theta_2}^2 + {\theta_3}^2)} = \frac{0.1}{1 + 0.5^2 + 0.4^2 + 0.1^2} = \mathbf{0.0704}$$

$$\rho(k) = \mathbf{0} \qquad \text{for } |k| > 3.$$

ACF: MA(q = 3)

Example (continuation): Plot of simulated series and ACF with 95% CI: = [**-0.1386, 0.1386**]

> sim_ma3_05 <- arima.sim(list(order=c(0,0,3), ma=c(0.5, 0.4, 0.2)), n=200)   # sim MA(3)



ACF: ARMA(1, 1)

Example: Sample ACF for an ARMA(1,1) process:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

• From the autocovariances, we get

$$\gamma(0) = \sigma^2 \frac{1 + \theta_1^{\,2} + 2\phi_1\theta_1}{1 - \phi_1^{\,2}}$$

$$\gamma(1) = \sigma^2 \frac{(1 + \phi_1\theta_1) * (\phi_1 + \theta_1)}{1 - \phi_1^{\,2}}$$

$$\gamma(k) = \phi_1 \gamma(k-1) = \phi_1^{\,k-1} \sigma^2 \frac{(1 + \phi_1\theta_1) * (\phi_1 + \theta_1)}{1 - \phi_1^{\,2}}$$

• Then,

$$\rho(k) = \phi_1^{\,k-1} \frac{(1 + \phi_1\theta_1) * (\phi_1 + \theta_1)}{1 + \theta_1^{\,2} + 2\phi_1\theta_1}$$

⇒ If $|\phi_1|<1$, exponential decay. Similar pattern to AR(1).

ACF: ARMA(1, 1)

Example (continuation): Sample ACF for an ARMA(1,1) process:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

The ACF for an ARMA(1,1):

$$\rho(k) = \phi_1^{\,k-1} \frac{(1 + \phi_1\theta_1) * (\phi_1 + \theta_1)}{1 + \theta_1^{\,2} + 2\phi_1\theta_1}$$

• Suppose $\phi_1 = \mathbf{0.4}$, $\theta_1 = \mathbf{0.5}$. Then,

$$\rho(0) = \mathbf{1}$$

$$\rho(1) = \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5} = \mathbf{0.6545}$$

$$\rho(2) = 0.4 * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5} = \mathbf{0.2618}$$

$$\rho(3) = 0.4^2 * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5} = \mathbf{0.0233}$$

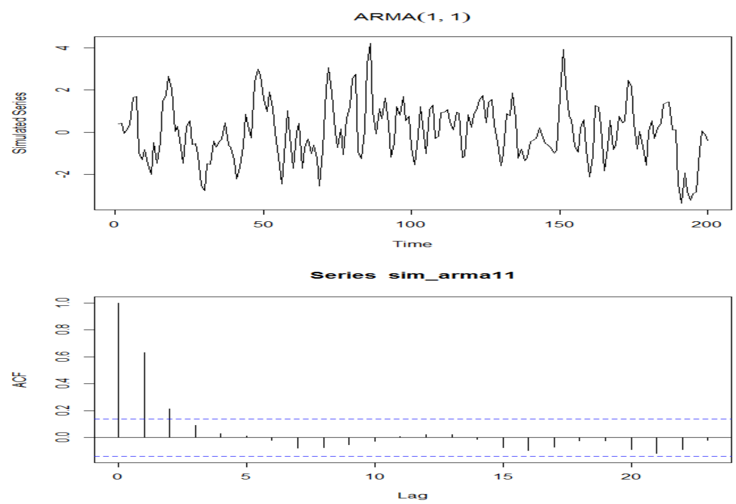
⋮

$$\rho(k) = 0.4^{k-1} * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5}$$

ACF: ARMA(1, 1)

Example (continuation): Plot of simulated series and ACF

> sim_arma11 <- arima.sim(list(order=c(1,0,1), ar=**0.4**, ma=**0.5**), n=200)   #sim ARMA(1,1)



ACF: Example – U.S. Stock Returns

Example: **US Monthly Returns** (1871 – 2020, T=1,795)

```
Sh_da <- read.csv("C://Financial Econometrics/Shiller_2020data.csv", head=TRUE,
sep=",")
x_P <- Sh_da$P
x_D <- Sh_da$D
T <- length(x_P)
lr_p <- log(x_P[-1]/x_P[-T])
lr_d <- log(x_D[-1]/x_D[-T])
acf_p <- acf(lr_p)                    # acf: R function that estimates the ACF
> acf_p
```

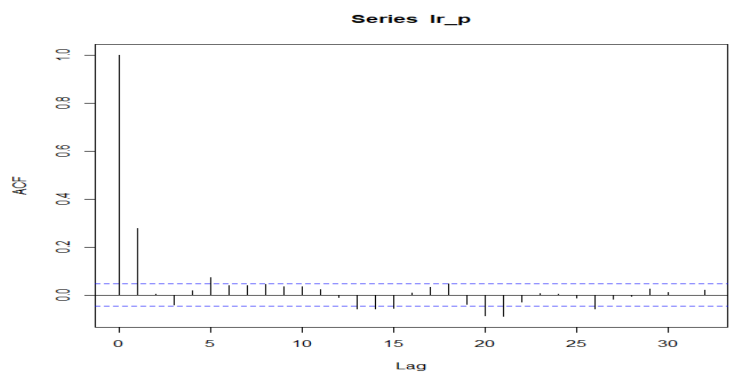
Autocorrelations of series 'lr_p', by lag

0	1	2	3	4	5	6	7	8	9	10	11
1.000	0.279	0.004	-0.043	0.017	0.074	0.039	0.039	0.044	0.035	0.034	0.022
12	13	14	15	16	17	18	19	20	21	22	23
-0.010	-0.059	-0.058	-0.056	0.009	0.033	0.047	-0.040	-0.087	-0.090	-0.029	0.005
24	25	26	27	28	29	30	31	32			
0.003	-0.013	-0.058	-0.018	-0.005	0.026	0.011	0.000	0.020			

SE(r_k) = 1/sqrt(T) = 1/sqrt(1,795) **= .0236** $\Rightarrow$ 95% CI: $\pm$ 2 * **0.0236**

ACF: Example – U.S. Stock Returns

Example (continuation): Correlogram for **US Monthly Returns** (1871 – 2020)



Note: With the exception of first correlation, correlations are small. However, many are significant, not strange result when T is large.

ACF: Example – U.S. Stock Returns

Example: **US Monthly Changes in Dividends** (1871 – 2020, T = 1,795)

```
acf_d <- acf(lr_d)
> acf_d
```

Autocorrelations of series 'lr_d', by lag

0	1	2	3	4	5	6	7	8	9	10	11
1.000	0.462	0.516	0.432	0.444	0.326	0.442	0.288	0.283	0.265	0.202	0.168

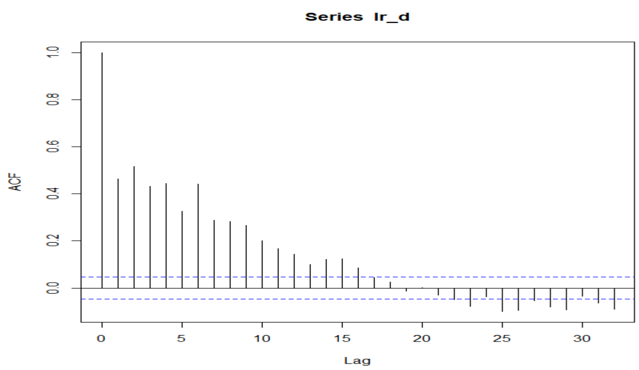
12	13	14	15	16	17	18	19	20	21	22	23
0.142	0.100	0.122	0.123	0.085	0.045	0.026	-0.013	0.001	-0.029	-0.049	-0.077

24	25	26	27	28	29	30	31	32
-0.038	-0.100	-0.095	-0.055	-0.081	-0.092	-0.034	-0.063	-0.089

High correlations and significant even after 32 months!

ACF: Example – U.S. Dividends

Example (continuation): Correlogram for **US Monthly Changes in Dividends** (1871 – 2020)



Note: Correlations are positive for almost 1.5 years, then become negative.

ACF: Joint Significance Tests

• Recall the Q statistic as:

$$Q = T \sum_{k=1}^{m} \hat{\rho}_k^2$$

Under H_0: $\rho_1 = \rho_2 = ... = \rho_m = 0$, Q follows χ_m^2

$$Q = T \sum_{k=1}^{m} \hat{\rho}_k^2 \xrightarrow{d} \chi_m^2$$

• The Ljung-Box (LB) statistic has better finite sample properties than the Q statistic. Under H_0, LB follows a χ_m^2:

$$LB = T(T+2) \sum_{k=1}^{m} \left(\frac{\hat{\rho}_k^2}{(T-k)}\right) \xrightarrow{d} \chi_m^2$$

ACF – Joint Significance Tests

Example: LB test with **20 lags** for **US Monthly Returns** and **Changes in Dividends** (1871 – 2020)

```
> Box.test(lr_p, lag=20, type= "Ljung-Box")
data:  lr_p
X-squared = 208.02, df = 20, p-value < 2.2e-16
```
⇒ Reject H_0 at 5% level.

```
> Box.test(lr_d, lag=20, type= "Ljung-Box")
data:  lr_d
X-squared = 2762.7, df = 20, p-value < 2.2e-16
```
⇒ Reject H_0 at 5% level.

Conclusion: We found joint significance of first 20 autocorrelations.

Partial ACF (PACF)

- The ACF gives us a lot of information about the order of the dependence when the series we analyze follows a MA process: The ACF is zero after q lags for an MA(q) process.

- If the series we analyze, however, follows an ARMA or AR, the ACF alone tells us little about the orders of dependence: We only observe an exponential decay.

- We introduce a new function that behaves like the ACF of MA models, but for AR models, namely, the partial autocorrelation function (PACF).

- The PACF is similar to the ACF. It measures correlation between observations that are k time periods apart, after controlling for correlations at intermediate lags.

Partial ACF

Intuition: Suppose we have an AR(1):

$$y_t = \phi_1\ y_{t-1} + \varepsilon_t.$$

Then,

$$\rho(2) = \phi_1{}^2$$

The correlation between y_t and y_{t-2} is not zero, as it would be for an MA(1), because y_t is dependent on y_{t-2} through y_{t-1}.

Suppose we break this chain of dependence by removing ("partialing out") the effect y_{t-1}. Then, we consider the correlation between $[y_t - \phi_1 y_{t-1}]$ & $[y_{t-2} - \phi_1 y_{t-1}]$ –i.e., the correlation between y_t & y_{t-2} with the linear dependence of each on y_{t-1} removed:

$\gamma(2)$ = Cov($y_t - \phi_1 y_{t-1}$, $y_{t-2} - \phi_1\ y_{t-1}$) = Cov($\varepsilon_t$, $y_{t-2} - \phi_1\ y_{t-1}$) =0

Similarly,

$\gamma(k)$ = Cov(ε_t, $y_{t-k} - \phi_1 y_{t-1}$) = 0 for all k > 1.

Partial ACF

Definition: The **PACF** of a stationary time series $\{y_t\}$ is ϕ_{hh}:

ϕ_{11}= Corr(y_t, y_{t-1}) = ρ(1)

ϕ_{hh} = Corr(y_t – E[$y_t\,|\,I_{t-1}$], y_{t-h} – E[$y_{t-h}\,|\,I_{t-1}$]) for h = 2, 3, ...

This removes the linear effects of y_{t-2}, ..., y_{t-h}.

Example: AR(p) process:

$$y_t = \mu + \phi_1\ y_{t-1} + \phi_2\ y_{t-2} + \ldots + \phi_p\ y_{t-p} + \varepsilon_t$$
$$E[y_t|I_{t-1}] = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_h y_{t-h-1}$$
$$E[y_{t-h}|I_{t-1}] = \mu + \phi_1 y_{t-h-1} + \phi_2 y_{t-h-2} + \ldots + \phi_h y_{t-1}$$

Then, ϕ_{hh} $= \phi_h$ if $1 \le h \le p$
$= 0$ otherwise

⇒ After the p^{th} *PACF,* all remaining PACF are 0 for AR(p) processes.

Partial ACF

• The PACF ϕ_{hh} is also the last coefficient in the **best linear prediction** of y_t given $y_{t-1}, y_{t-2}, ..., y_{t-h}$. (⇒ OLS!)

OLS estimation steps:
Regress y_t against y_{t-1} ⇒ ϕ_{11}: estimated coefficient of y_{t-1}.

Regress y_t against y_{t-1} & y_{t-2} ⇒ ϕ_{22}: estimated coefficient of y_{t-2}.
⋮
Regress y_t against y_{t-1}, y_{t-2}, … y_{t-h}⇒ ϕ_{hh}: estimated coefficient of y_{t-h}.

• OLS estimation is simple, easy to use. Estimation by Yule-Walker equation is possible. The is also a recursive algorithm by Durbin-Levinson.

• The plot of the PACF is called the **partial correlogram**.

Inverse ACF (IACF)

• The IACF of the ARMA(p, q) model

$$\phi(L)\, y_t = \theta(L)\varepsilon_t$$

is defined to be (assuming invertibility) the ACF of the *inverse* (or *dual*) process

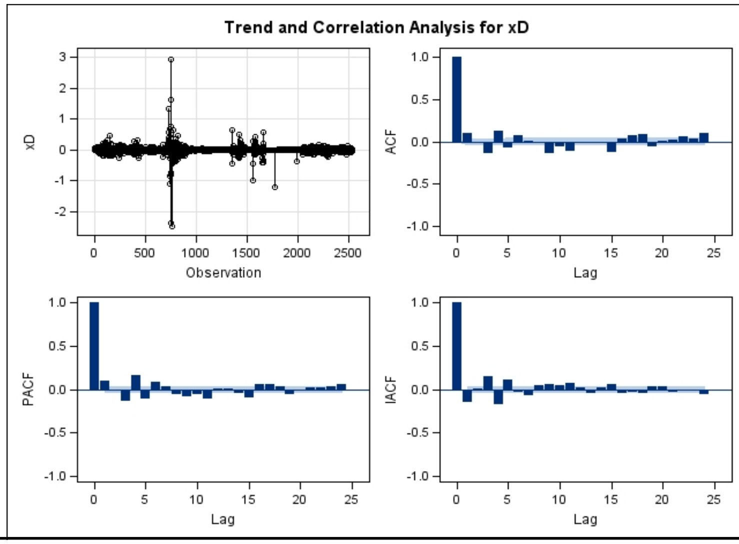
$$\theta(L)\, {y_t}^{-1} = \phi(L)\, \varepsilon_t$$

• The IACF has the same property as the PACF: AR(p) is characterized by an IACF that is nonzero at lag p but zero for larger lags.

• The IACF can also be used to detect over-differencing. If the data come from a nonstationary or nearly nonstationary model, the IACF has the characteristics of a noninvertible moving-average.

ACF, Partial ACF & IACF: Example

Example: Monthly USD/GBP 1st differences (1800-2013)



Non-Stationary Time Series Models

• A trend is usually easy to spot. A more sophisticated visual tool is the ACF: a slow decay in ACF is indicative of highly correlated data, which suggests a trend.

• A series with a trend is not stationary. To build a forecasting model, we need to remove the trend from the series. The models we consider:

(1) Deterministic trend: y_t is a function of t. For example,

$$y_t = \alpha + \beta\, t + \varepsilon_t$$

(2) Stochastic trend: y_t is a function of aggregated errors, ε_t, over time. For example,

$$y_t = \mu + y_{t-1} + \varepsilon_t = y_0 + t\,\mu + \sum_{j=0}^{t} \varepsilon_{t-j}$$

• The process to remove the trend depends on the structure of the DGP of y_t.

Non-Stationary Time Series Models

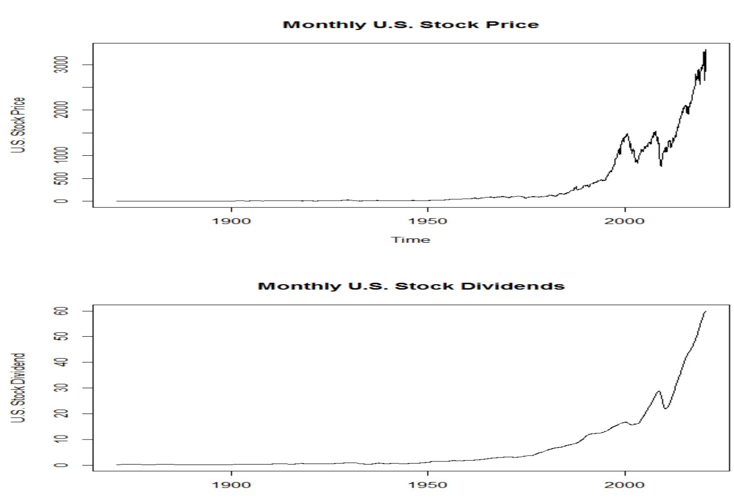
• The process to remove the trend depends on the nature of the DGP of the trending y_t:

(1) Deterministic trend – Simple model: $y_t = \alpha + \beta\, t + \varepsilon_t$
– Solution: **Detrending** –i.e., regress y_t on a constant and a time trend, t. Then, keep residuals for further modeling.

(2) Stochastic trend – Simple model: $y_t = \mu + y_{t-1} + \varepsilon_t$.
– Solution: **Differencing** –i.e., apply $\Delta = (1 - L)$ operator to y_t. Then, use Δy_t for further modeling.

Non-Stationary Time Series Models

Example: Plot of US Monthly Prices and Dividends (1871 – 2020)



Non-Stationary Models: Deterministic Trend

• Suppose we have the following model, with a determinist trend:

$$y_t = \alpha + \beta\, t + \varepsilon_t.$$

• $\{y_t\}$ will show only temporary departures from trend line $\alpha + \beta\, t$. It is a model with short memory. A shock (big ε_t) hits y_t, y_t goes back to trend level in short time. Forecasts are not affected.

• This type of model is called a **trend stationary** (TS) model.

• Note that trivially, by definition, ε_t is WN. Then, removing $\alpha + \beta\, t$ from y_t creates a WN series –i.e., the influence of t from y_t is gone:

$$\varepsilon_t = y_t - \alpha - \beta\, t$$

• When we replace α & β by their OLS estimates, we **detrend** y_t. The residual from the OLS is called **detrended** y_t.

$$e_t = y_t - \hat{\alpha} - \hat{\beta}\, t$$ (the residuals are the *detrended* y_t series)

Non-Stationary Models: Deterministic Trend

• We can detrend in more complicated models. For example, suppose e have a stationary AR(p) model with linear and quadratic trends:

$$y_t = \alpha + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \beta_1 t + \beta_2 t^2 + \varepsilon_t.$$

• Note that removing from y_t a constant, a linear and a quadratic trend creates a series, w_t, which is composed of a WN error, ε_t, and the AR(p) part:

$$w_t = \varepsilon_t + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} = y_t - \alpha - \beta_1 t - \beta_2 t^2$$

• This is a stationary series: the dependence on t is gone. We will work with the residual from a regression of y_t against a constant, t and t^2:

$$\hat{w}_t = y_t - \hat{\alpha} - \hat{\beta}_1\, t - \hat{\beta}_2 t^2$$ ($\hat{w}_t$= **detrended** y_t).

Remark: We do not necessarily get stationary series by detrending.

Non-Stationary Models: Deterministic Trend

• Many economic series exhibit "exponential trend/growth". They grow over time like an exponential function over time instead of a linear function. In this cases, it is common to work with logs

$$\ln(y_t) = \alpha + \beta\, t + \varepsilon_t. \qquad (\Rightarrow y_t = e^{\alpha + \beta\, t + \varepsilon_t})$$

$\Rightarrow$ The average growth rate is: $E[\Delta \ln(y_t)] = \beta$

• We can have a more general model:

$$y_t = \alpha + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \beta_1 t + \beta_2 t^2 + \ldots + \beta_k t^k + \varepsilon_t.$$

• Estimation of AR(p) with a trend component:
- **OLS**.
- **Frish-Waugh method** (a 2-step method):
  (1) Detrend y_t: regress y_t against a constant & a time trend, t. Then, get the residuals (=y_t without the influence of t).
  (2) Use residuals to estimate the AR(p) model.

Non-Stationary Models: Deterministic Trend

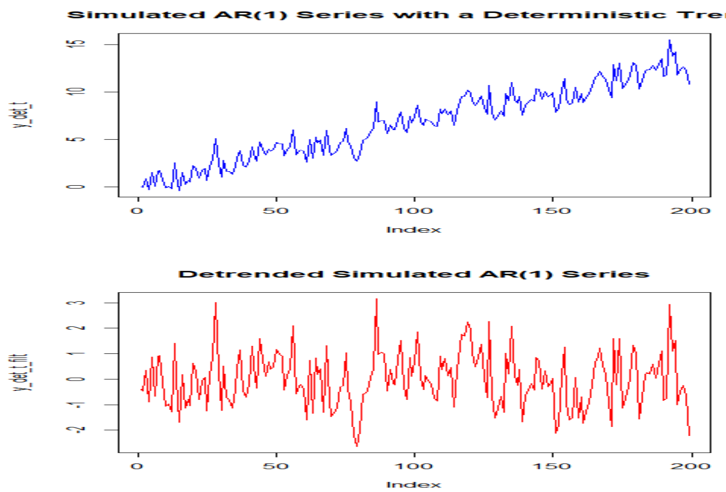
Simulated Example: We simulate an AR(1) series with a trend:

$$y_t = 0.3 + 0.2\, y_{t-1} + 0.05\, t + \varepsilon_t.$$

```
T_sim <- 200                                      # Length of simulation
y_sim <- matrix(0,T_sim,1)                        # Vector to accumulate simulated data
u <- rnorm(T_sim, sd = 1)                         # Draw T_sim normally distributed errors
mu <- 0.3                                         # Constant
phi1 <- 0.2                                       # Change to create different AR(1) patterns
mu_t <- .05                                       # Trend coefficient
t <- 2                                            # Time index for observations
while (t <= T_sim) {
 y_sim[t] = mu + phi1 * y_sim[t-1] +  mu_t * t + u[t]  # y_sim simulated values
  t <- t + 1
}
y_det_t <- y_sim[2: T_sim]
plot(y_det_t, type="l", col = "blue", main = "Simulated Series with a Deterministic Trend")

# Detrend series
trend <- c(1:(T_sim1-1))
fit_det_t <- lm(y_det_t ~ trend)
y_det_t_filt <- fit_det_t$residuals                          # Filtered series
plot(y_det_t_filt, type="l", main = "Detrended Simulated Series")
```

Non-Stationary Models: Deterministic Trend

Simulated Example (continuation): We plot the simulated AR(1) series (blue) and the detrended simulated series (red).



Non-Stationary Models: Deterministic Trend

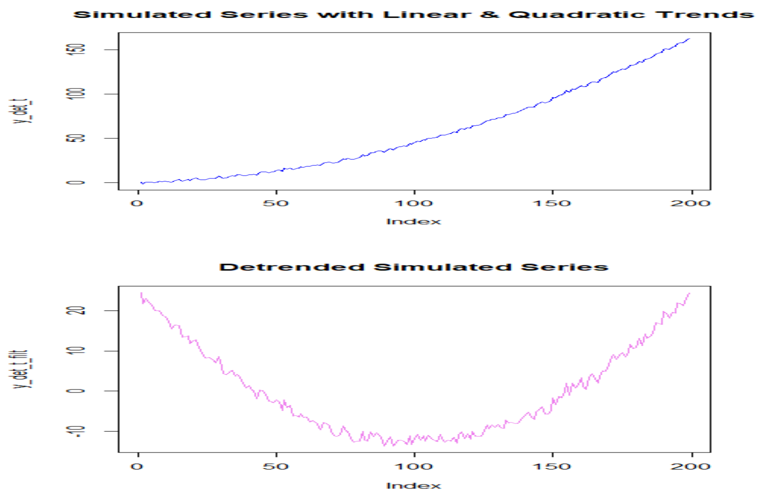
Simulated Example (continuation): Now, we add a quadratic trend:

$$y_t = 0.3 + 0.2\, y_{t-1} + 0.05\, t + 0.003\, t^2 + \varepsilon_t.$$

```
mu_t2 <- .003                                      # Trend square coefficient
t <- 2                                             # Time index for observations
while (t <= T_sim) {
y_sim[t] = mu + phi1 * y_sim[t-1] +  mu_t * t + u[t]   # y_sim simulated autocorrelated values
t <- t + 1
}
y_det_t <- y_sim[2: T_sim]
plot(y_det_t, type="l", col = "blue", main = "Simulated Series with a Deterministic Trend")

# Detrend series with only a linear trend
trend <- c(1:(T_sim1-1))
fit_det_t <- lm(y_det_t ~ trend)
y_det_t_filt <- fit_det_t$residuals                # Filtered series
plot(y_det_t_filt, type="l", main = "Detrended Simulated Series")

## Detrend series with a linear & Quadratic trends
trend2 <- trend^2fit_det_t <- lm(y_det_t ~ trend + trend2)
y_det_t_filt <- fit_det_t$residuals                # Filtered series
plot(y_det_t_filt, type="l", col = "violet", main="Detrended Simulated Series")
```

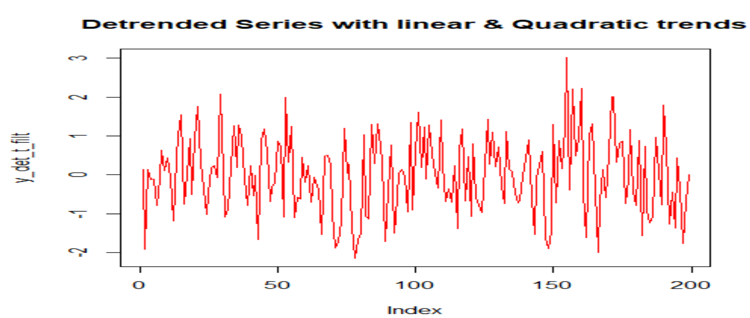
Non-Stationary Models: Deterministic Trend

Simulated Example (continuation): We plot the simulated AR(1) series (blue) and the detrended series with a linear trend (violet).



Non-Stationary Models: Deterministic Trend

Simulated Example (continuation): We plot the detrended simulated series with a linear and quadratic trends (red).

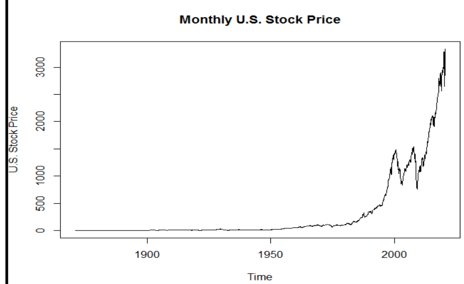


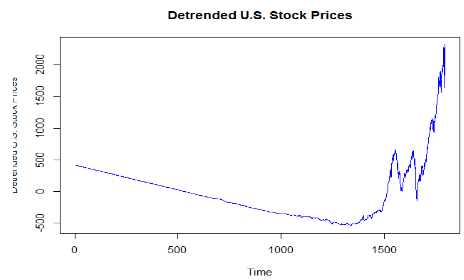
Remark: A series with a quadratic trend, needs to be detrended with a quadratic trend, otherwise extra patterns (U-shape, in this case) appear. Once we use an appropriate detrending model, we use the detrended series –i.e., the residuals– for furthering (ARMA) modeling.

Non-Stationary Models: Deterministic Trend

Example: We detrend U.S. Stock Prices

```
T <- length(x_P)                         # length of series
trend <- c(1:T)                          # create trend
det_P <- lm(x_P ~ trend)                 # regression to get detrended e
detrend_P <- det_P$residuals
plot(detrend_P, type="l", col="blue", ylab ="Detrended U.S. Prices", xlab ="Time")
title("Detrended U.S. Stock Prices")
```



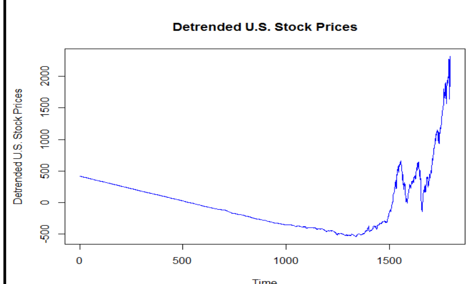


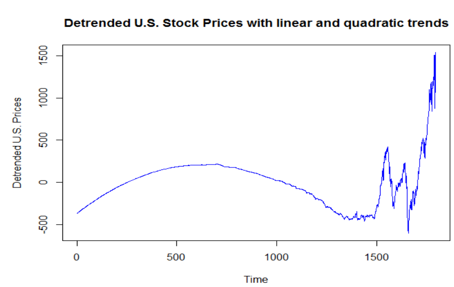
Note: Extra pattern in detrended series ⇒ Using the wrong model.

Non-Stationary Models: Deterministic Trend

Example: We detrend U.S. Stock Prices adding a square trend

```
trend2 <- trend^2
det_P <- lm(x_P ~ trend + trend2)        # regression to get detrended e
detrend_P <- det_P$residuals
plot(detrend_P, type="l", col="blue", ylab ="Detrended U.S. Prices", xlab ="Time")
title("Detrended U.S. Stock Prices with linear and quadratic trends")
```



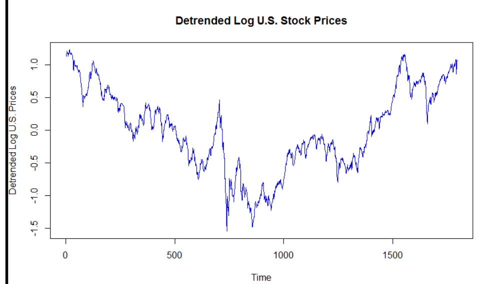


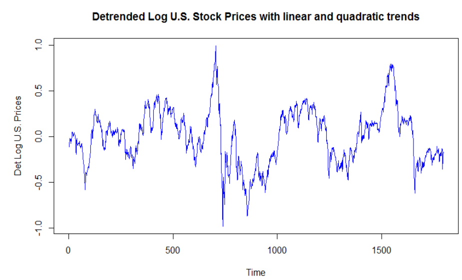
⇒ Still using the wrong model to detrend: Try exponential trend.

Non-Stationary Models: Deterministic Trend

Example: We detrend Log U.S. Stock Prices adding a squared trend

```
l_P <- log(x_P)
det_lP <- lm(l_P ~ trend)                              # regression to get detrended e
detrend_lP <- det_lP$residuals
plot(detrend_lP, type="l", col="blue", ylab ="Detrended Log U.S. Prices", xlab ="Time")
title("Detrended Log U.S. Stock Prices")

det_lP2 <- lm(l_P ~ trend + trend2)                    # regression to get detrended e
det_lP2 <- det_lP2$residuals
plot(det_lP2, type="l", col="blue", ylab ="Det Log U.S. Prices", xlab ="Time")
title("Detrended Log U.S. Stock Prices with linear and quadratic trends")
```





Non-Stationary Models: Stochastic Trend

• The more modern approach is to consider trends in time series as a variable trend.

• A variable trend exists when a trend changes in an unpredictable way. Therefore, it is considered **stochastic**.

• Recall the AR(1) model: $y_t = \mu + \phi_1 \, y_{t-1} + \varepsilon_t$

• As long as $|\phi_1| < 1$, everything is fine, we have a stationary AR(1) process: OLS is consistent, t-stats are asymptotically normal, etc.

• Now consider the special case where $\phi_1 = 1$:

$$y_t = \mu + y_{t-1} + \varepsilon_t$$

Q: Where is the (stochastic) trend? No t term.

Non-Stationary Models: Stochastic Trend

- Let us replace recursively the lag of y_t on the right-hand side:

$$\begin{aligned} y_t &= \mu + y_{t-1} + \varepsilon_t \\ &= \mu + (\mu + y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &\dots \\ &= y_0 + t\,\mu + \textstyle\sum_{j=0}^{t} \varepsilon_{t-j} \end{aligned}$$

Deterministic trend | Accumulation of errors (shocks) – stochastic part

- This process is called a **Random walk with drift**: y_t grows with t.

- Each ε_t shock represents a shift in the intercept. All values of $\{\varepsilon_t\}$ have a 1 as coefficient $\Rightarrow$ each shock never vanishes (permanent).

- We remove the trend by **differencing** y_t

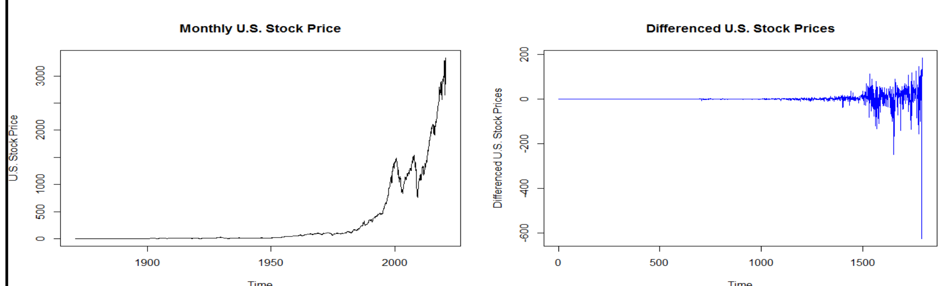
$$\Rightarrow \Delta y_t = (1 - L)\, y_t = \mu + \varepsilon_t$$

Note: Applying the $(1 - L)$ operator to a time series is called *differencing*

Non-Stationary Models: Stochastic Trend

Example: We difference U.S. Stock Prices, using the *diff* R function:

```
diff_P <- diff(x_P)
> plot(diff_P,type="l", col="blue", ylab ="Differenced U.S. Stock Prices", xlab ="Time")
> title("Differenced U.S. Stock Prices")
```



Remark: Trend is gone $\Rightarrow$ Use first differences for AR modeling.

Non-Stationary Models: Stochastic Trend

• y_t is said to have a *stochastic trend* (ST), since each ε_t shock gives a permanent and random change in the conditional mean of the series.

• For these situations, we use **Autoregressive Integrated Moving Average** (**ARIMA**) models.

• Q: Deterministic or Stochastic Trend?
They appear similar: Both lead to growth over time. The difference is how we think of ε_t. Should a shock today affect y_{t+1}?

– TS: $y_{t+1} = \mu + \beta\,(t+1) + \varepsilon_{t+1}$ $\Rightarrow \varepsilon_t$ does not affect y_{t+1}.

– ST: $y_{t+1} = \mu + y_t + \varepsilon_{t+1} = \mu + [\mu + y_{t-1} + \varepsilon_t] + \varepsilon_{t+1}$
$= 2 * \mu + y_{t-1} + \varepsilon_t + \varepsilon_{t+1}$ $\Rightarrow \varepsilon_t$ affects y_{t+1}.
(In fact, the shock ε_t has a *permanent* impact.)

ARIMA(p, d, q) Models

• For $p, d, q \geq 0$, we say that a time series $\{y_t\}$ is an *ARIMA* (p, d, q) *process* if $w_t = \Delta^d\, y_t = (1-L)^d\, y_t$ is ARMA(p, q). That is,

$$\phi(L)(1-L)^d\; y_t = \theta(L)\,\varepsilon_t$$

• Applying the $(1-L)$ operator to a time series is called *differencing*.

Notation: If y_t is non-stationary, but $\Delta^d y_t$ is stationary, then y_t is *integrated* of order d, or I(d). A time series with *unit root* is I(1). A stationary time series is I(0).

Examples:
Example 1: RW: $y_t = y_{t-1} + \varepsilon_t$.
y_t is non-stationary, but

$w_t = (1-L)\; y_t = \varepsilon_t$ $\Rightarrow w_t \sim$ WN!

Now, $y_t \sim$ ARIMA(0, 1, 0).

ARIMA(p, d, q) Models

Example 2: AR(1) with time trend: $y_t = \mu + \delta\, t + \phi_1\, y_{t-1} + \varepsilon_t$.
y_t is non-stationary, but

$$w_t = (1 - L)\, y_t$$
$$= \mu + \delta\, t + \phi_1 y_{t-1} + \varepsilon_t - [\mu + \delta\,(t - 1) + \phi_1\, y_{t-2} + \varepsilon_{t-1}].$$
$$= \delta + \phi_1 w_{t-1} + \varepsilon_t - \varepsilon_{t-1} \qquad \Rightarrow w_t \sim \text{ARMA}(1, 1).$$

Now, $y_t \sim$ ARIMA(1, 1, 1).

• We call both process *first difference stationary*.

Note:
– Example 1: Differencing a series with a unit root in the AR part of the model reduces the AR order.

– Example 2: Differencing can introduce an extra MA structure. We introduced non-invertibility (θ_1= 1). This happens when we difference a TS series. Detrending should be used in these cases.

ARIMA(p, d, q) Models

• In practice:
A root near 1 of the AR polynomial ⇒ differencing
A root near 1 of the MA polynomial ⇒ over-differencing

• In general, we have the following results:
- Too little differencing: not stationary.
- Too much differencing: extra dependence introduced.

• Finding the right d is crucial. For identifying preliminary values of d:
- Use a time plot.
- Check for slowly decaying (persistent) ACF/PACF.

Note: There are many formal tests for unit roots. Most popular tests: ADF (Augmented Dickey-Fuller) and PP (Phillips-Perron).

ARIMA Models: Unit Roots 1?

Example 1: Monthly Stock Price levels (1871-2020)

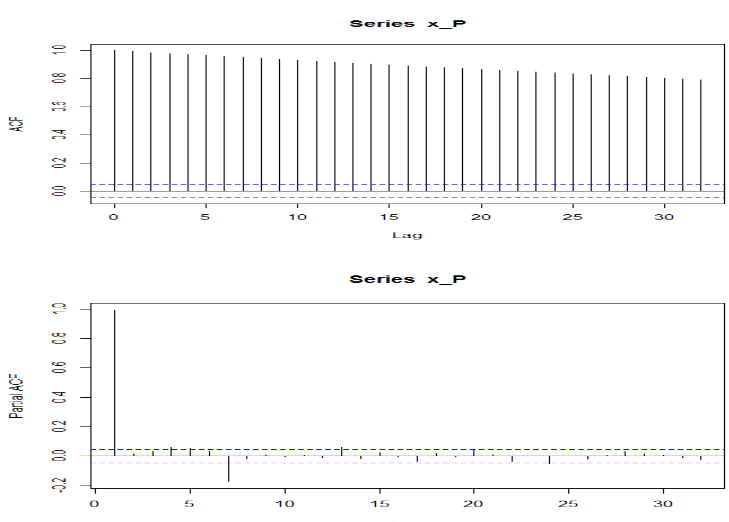
acf_P <- acf(x_P)
> acf_P
Autocorrelations of series 'x_p', by lag

0	1	2	3	4	5	6	7	8	9	10	11
1.000	0.992	0.984	0.977	0.971	0.966	0.961	0.954	0.946	0.938	0.931	0.924
12	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
0.917	0.911	0.904	0.897	0.891	0.884	0.877	0.871	0.865	0.860	0.854	0.848
24	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**			
0.841	0.834	0.827	0.821	0.815	0.809	0.803	0.797	0.790			

Very high autocorrelations. Looks like $\phi_1 \approx 1$.

ARIMA Models – Unit Roots 1: ACF & PACF

Example 1: Monthly Stock Price levels (1871-2020)



ARIMA Models: Unit Roots 2?

Example 2: Monthly Interest Rates (1871-2020)

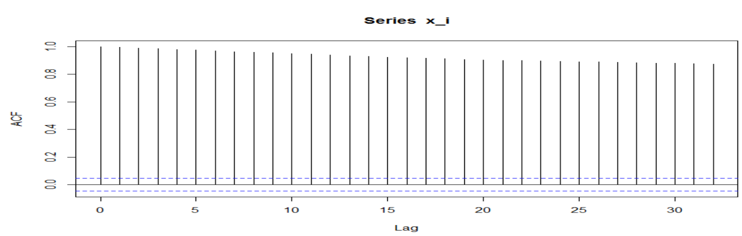
```
acf_i <- acf(x_i)
> acf_i
Autocorrelations of series 'x_i', by lag
```

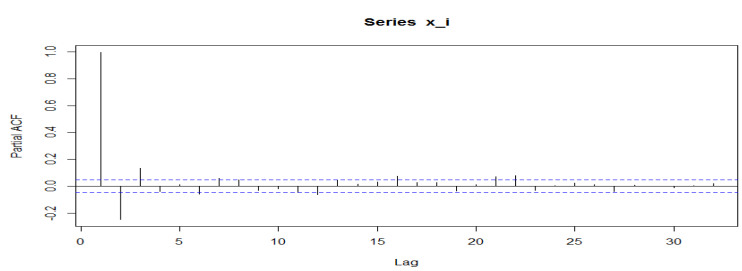
0	1	2	3	4	5	6	7	8	9	10	11
1.000	0.996	0.990	0.985	0.980	0.975	0.970	0.965	0.960	0.956	0.951	0.946
12	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
0.940	0.934	0.929	0.924	0.919	0.915	0.912	0.908	0.904	0.901	0.899	0.896
24	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**			
0.894	0.891	0.889	0.887	0.884	0.882	0.879	0.877	0.874			

Very high autocorrelations. Looks like $\phi_1 \approx 1$.

ARIMA Models – Unit Roots 2: ACF & PACF

Example 2: Monthly Interest Rates (1871-2020)





ARIMA Models – Random Walk

- A **random walk** (**RW**) is a process where the current value of a variable is composed of the past value plus an error term defined as a white noise (a normal variable with zero mean and variance one).

- RW is an ARIMA(0,1,0) process
$y_t = y_{t-1} + \varepsilon_t \quad \Rightarrow \Delta y_t = (1-L)y_t = \varepsilon_t, \qquad \varepsilon_t \sim WN(0, \sigma^2).$

- Popular model. Used to explain the behavior of financial assets, unpredictable movements (Brownian motions, drunk persons).

- A special case (limiting) of an AR(1) process: a **unit-root** process.

- Implication: E[$y_{t+1} \mid I_t$] = y_t $\quad \Rightarrow \Delta y_t$ is absolutely random.

- Thus, a RW is nonstationary, and its variance increases with t.

ARIMA Models – RW with Drift

- Change in y_t is partially deterministic (μ) and partially stochastic.
$$y_t - y_{t-1} = \Delta y_t = \mu + \varepsilon_t$$

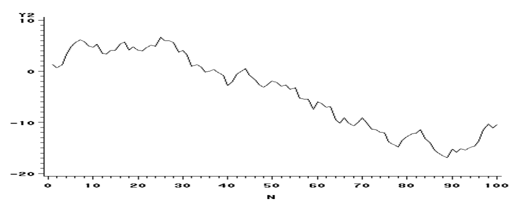
- Recall that y_t can also be written as
$$y_t = y_0 + t\,\mu \; + \textstyle\sum_{j=0}^{t} \varepsilon_{t-j}$$
$\Rightarrow \varepsilon_t$ has a permanent effect on the mean of y_t.

- Recall the difference between conditional and unconditional forecasts:
E[y_t] = $y_0 + t\,\mu$ (Unconditional forecast)
E[$y_{t+s} \mid y_t$] = $y_t + s\,\mu$ (Conditional forecast)

ARIMA Models – Random Walk

Examples: A simulated RW in R

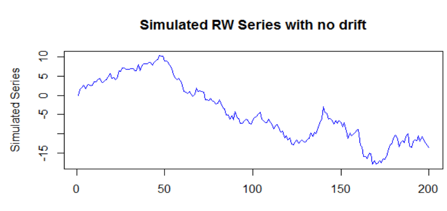
```
T_sim <- 200
u <- rnorm(200)              # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)
rho <- 1                     # Change to create different correlation patterns
a <- 2
mu <- 0                      # Time index for observations
while (a <= T_sim) {
        y_sim[a] = mu + rho * y_sim[a-1] + u[a] # y_sim simulated autocorrelated values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Simulated RW Series with no drift")
```

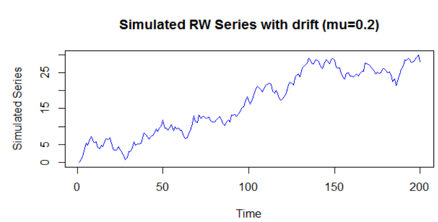


ARIMA Models – Random Walk

Examples: Two simulated RW one with drift and one without drift

```
T_sim <- 200                 # Sample size for simulation
u <- rnorm(200)              # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)   # Vector to collect simulated data
phi <- 1                     # Set phi = 1 for RW
a <- 2                       # Time index for observations
mu <- 0                      # RW Drift
while (a <= T_sim) {
        y_sim[a] = mu + phi * y_sim[a-1] + u[a]     # y_sim simulated RW values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Simulated RW Series with no drift")
```





ARIMA Models: Box-Jenkins

• We have a family of ARIMA models, indexed by p, q, and d.
Q: How do we select one?

An effective procedure for building empirical time series models is the Box-Jenkins approach, which consists of three stages:

(1) **Identification** or Model specification (order of ARIMA)

(2) **Estimation** of order p, q.

(3) **Diagnostics testing** on residuals:
$\Rightarrow$ Are they white noise? If not, add lags (p, q, or both).

If we are happy with model, then we proceed to **forecasting**.

ARIMA Models: Identification

• Recall the two main approaches to (1) Identification.
- **Correlation approach**: Based on ACF & PACF.
1) Make sure data is stationary –check a time plot. If not, differentiate.
2) Using ACF & PACF, guess small values for p & q.

- **Information criteria**: Very common situation: The order choice not clear from looking at ACF & PACF. Then, use AIC (or $AICc$), BIC, or HQIC (Hannan and Quinn (1979)).

This is the usual (& easier) approach.

R Note: The R function auto.arima uses $AICc$ to select p, q; d is selected using a formal unit root test (KPSS).

• Value parsimony. When in doubt, keep it simple (KISS).

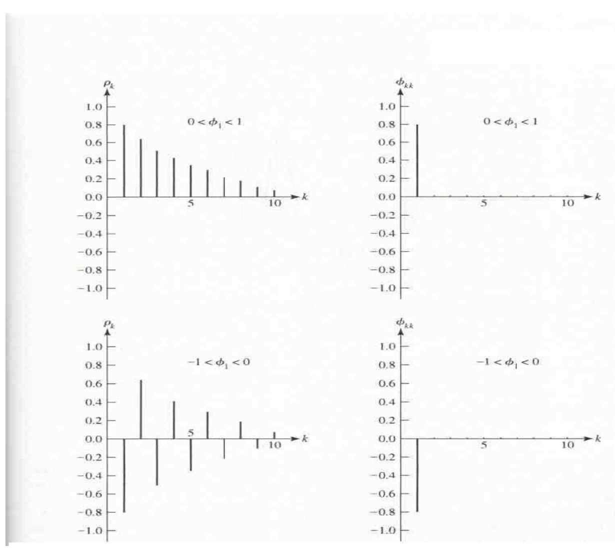
ARIMA Models: Identification – Correlations

• Correlation approach.
- ACF identifies order of MA: Non-zero at lag q; zero for lags > q.
  - PACF identifies order of AR: Non-zero at lag p; zero for lags > p.
  - All other cases, try ARMA(p, q) with p > 0 and q > 0.

Summary: For p > 0 and q > 0.

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	0 after lag q	Tails off
PACF	0 after lag p	Tails off	Tails off

ARIMA Models: Identification – AR(1)



64

ARIMA Models: Identification – AR(2)

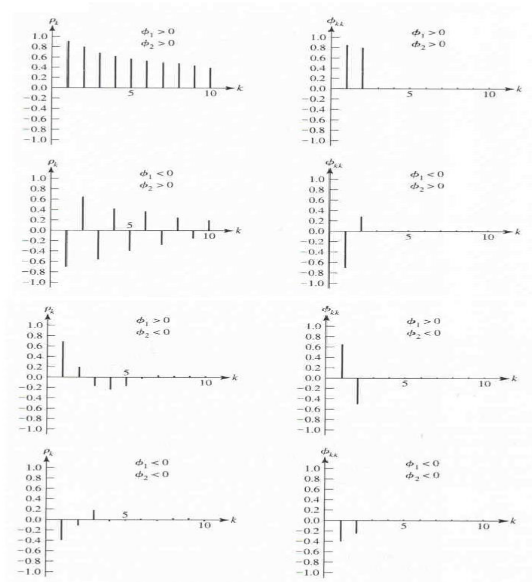


FIGURE ACF and PACF of AR(2) process: $(1 - \phi_1 B - \phi_2 B^2)\dot{Z}_t = a_t$.

65

ARIMA Models: Identification – MA(1)

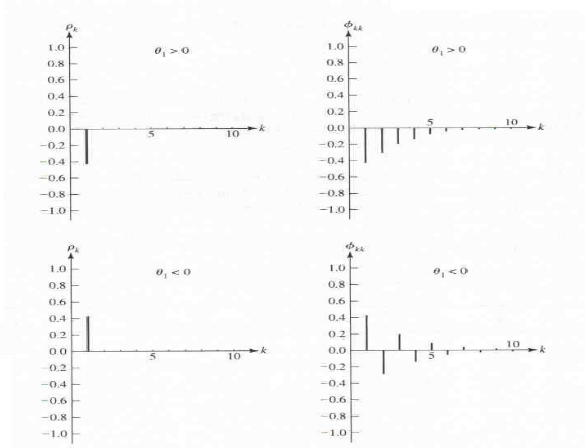
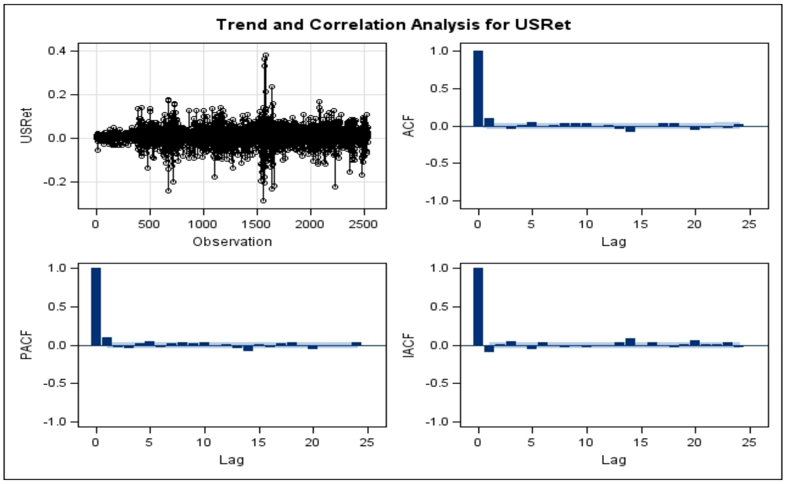


FIGURE ACF and PACF of MA(1) processes: $\dot{Z}_t = (1 - \theta_1 B)a_t$.

66

ARIMA Models: Identification – ARMA(1,1)

Example: Monthly US Returns (1800 - 2013).



• Note: Identification is not clear.

67

ARIMA Model: Identification – IC

• IC's are equal to the estimated variance or the log-likelihood function plus a penalty factor, that depends on k. Many IC's. Popular ones:

- **Akaike Information Criterion (AIC)**

$AIC = -2 * (lnL - k) = -2\ lnL + 2 * k$

$\Rightarrow$ if normality AIC = $T * ln(\frac{e'e}{T}) + 2 * k$ (+constants)

- **Bayes-Schwarz Information Criterion (BIC or SBIC)**

$BIC = -2 * lnL - ln(T) * k$

$\Rightarrow$ if normality AIC = $T * ln(\frac{e'e}{T}) + \ln(T) * k$ (+constants)

- **Hannan-Quinn (HQIC)**

HQIC = $-2 * (lnL - k\ [ln(ln(T))])$

$\Rightarrow$ if normality AIC = $T * ln\left(\frac{e'e}{T}\right) + 2 * k\ [ln(ln(T))]$ (+constants)

ARIMA Model: Identification – IC

• There are modifications of IC to get better finite sample behavior, a popular one is AIC corrected, $AICc$, statistic:

$$AICc = T\, ln\hat{\sigma}^2 + \frac{2k(k+1)}{T-k-1}$$

• $AICc$ converges to AIC as T gets large. Using $AICc$ is not a bad idea.

• For AR(p) models, other AR-specific criteria are possible: Akaike's final prediction error (FPE), Akaike's BIC, Parzen's CAT.

• **Hannan and Rissannen's** (1982) **minic** *(=Min*imum IC*)*: Calculate the BIC for different p*'s* (estimated first) and different q*'s*. Select the best model –i.e., lowest BIC.

Note: Box, Jenkins, and Reinsel (1994) proposed using the AIC above.

ARIMA Model: Identification – IC

• We would like the IC statistics –i.e., the IC's– to have good properties. For example, if the true model is being considered among many, we want the IC to select it. This can be done on average (unbiased) or as T increases (consistent).

Some results regarding AIC and BIC.

- AIC and Adjusted R^2 are **not consistent**.
- AIC is conservative –i.e., it tends to over-fit: k_{AIC} too large models.
- In time series, AIC selects the model that minimizes the out-of-sample one-step ahead forecast MSE.
- BIC is **more parsimonious** than AIC. It penalizes the inclusion of parameters more ($k_{BIC} \leq k_{AIC}$).
- BIC is **consistent** in autoregressive models.
- No agreement which criteria is better.

ARIMA Model: Identification – IC

• **Example**: Monthly US Returns (1800 - 2013) Hannan and Rissannen (1982)'s minic.

Minimum Information Criterion

Lags	**MA 0**	**MA 1**	**MA 2**	**MA 3**	**MA 4**	**MA 5**
AR 0	-6.1889	**-6.19573**	-6.19273	-6.19177	-6.18872	-6.18886
AR 1	-6.19511	-6.193	-6.19001	-6.18929	-6.18632	-6.18678
AR 2	-6.19271	-6.18993	-6.1911	-6.18802	-6.18536	-6.1839
AR 3	-6.19121	-6.18916	-6.18801	-6.18562	-6.18256	-6.18082
AR 4	-6.18853	-6.18609	-6.18523	-6.18254	-6.17983	-6.17774
AR 5	-6.18794	-6.18671	-6.18408	-6.18099	-6.1779	-6.17564

• Note: Best Model is ARMA(0, 1).

ARIMA Model: Identification – IC

• Script in R to select model using *arima* function.

```
p <- 6                                          # set max order for AR part: p-1
q <- 6                                          # set max order for Ma part: q-1
npq <- p*q
aic_m <- matrix(0,nrow = npq, ncol=3)           # matrix collects p, q, AIC: AIC in last column
j <- 0
k <- 1
while (j < p) {
i <- 0
while (i < q) {
mod_j <- arima(lr_p, order=c(i,0,j))            # fit arima(p,0,q) process
aic_m[k,] <- cbind(i, j, mod_j$aic)             # extract aic from arima fit model
i <- i + 1
k <- k + 1
}
j <- j + 1
}
aic_m                                           # Print all the results AR(i), MA(j), AIC
min_aic <- min(aic_m[,3])                       # Minimum AIC
min_aic                                         # Print Minimum

which(aic_m == min_aic, arr.ind=TRUE)           # Prints the row
```

ARIMA Model: Identification – IC

Example: Monthly US Returns (1871 - 2020).
R has a couple of functions that select automatically the "best" ARIMA model: *armaselect* (using package *auto*) minimizes BIC and *auto.arima* (using package *forecast*) minimizes AIC, $\boldsymbol{AICc}$ (default) or BIC.

```
> armaselect(lr_p)                    # shows the best 10 models according to BIC
      p q       sbc
 [1,] 2 0 -11644.79
 [2,] 1 0 -11641.53
 [3,] 3 0 -11637.71
 [4,] 4 0 -11632.43
 [5,] 5 0 -11629.95
 [6,] 2 1 -11627.42
 [7,] 6 0 -11621.70
 [8,] 1 3 -11620.18
 [9,] 3 1 -11619.93
[10,] 2 2 -11619.44
```

ARIMA Model: Identification – IC

Example: Monthly US Returns (1871 - 2020).

```
> auto.arima(lr_p, ic="bic", trace=TRUE)                # ic="BIC". function
approximates models.

 Fitting models using approximations to speed things up...

 ARIMA(2,0,2) with non-zero mean : -6519.957
 ARIMA(0,0,0) with non-zero mean : -6392.599
 ARIMA(1,0,0) with non-zero mean : -6527.879
 ARIMA(0,0,1) with non-zero mean : -6536.548
 ARIMA(0,0,0) with zero mean     : -6385.246
 ARIMA(1,0,1) with non-zero mean : -6529.358
 ARIMA(0,0,2) with non-zero mean : -6530.806
 ARIMA(1,0,2) with non-zero mean : -6523.415
 ARIMA(0,0,1) with zero mean     : -6534.284

 Now re-fitting the best model(s) without approximations...

 ARIMA(0,0,1) with non-zero mean : -6536.463
```


ARIMA Model: Identification – IC

Example (continuation): Monthly US Returns (1871 - 2020).

```
> auto.arima(lr_p, ic="bic", max.p=5, max.q = 5, trace=TRUE)          # approximates models.

Series: lr_p
ARIMA(0,0,1) with non-zero mean

Coefficients:
         ma1    mean
      0.2880  0.0037
s.e.  0.0218  0.0012

sigma^2 estimated as 0.001523:  log likelihood=3279.47
AIC=-6552.94   AICc=-6552.93   BIC=-6536.46
```

• auto.arima does not try a lot of models, tries to keep the $p + q \leq 5$.

Remark: Do not take the results from auto.arima or armaselect or minic as the final model. We still need to check the residuals are WN.

ARIMA Model: Identification – IC - Remarks

• The old correlation approach is seldom used. In practice, identification is done with ICs.

• There is no agreement on which IC is best. The AIC is the most popular, but others are also used.

• Asymptotically, the **BIC is consistent** –i.e., it selects the true model if, among other assumptions, the true model is among the candidate models considered.

• The AIC is not consistent, generally producing too large a model, but **is more efficient** –i.e., when the true model is not in the candidate model set, the AIC asymptotically chooses whichever model minimizes the MSE/MSPE.

ARIMA Process – Estimation

• We assume:
- The model order $d,\ p$, and q is known. Make sure y_t is I(0).
- The data has zero mean (μ=0). If this is not reasonable, demean y_t.

Fit a zero-mean ARMA model to the demeaned y_t:

$$\phi(L)(y_t - \bar{y}) = \theta(L)\varepsilon_t$$

• Several ways to estimate an ARMA(p, q) model:

1) ***Maximun Likelihood Esimation* (MLE)**. Assume a distribution, usually a normal distribution, and, then, do ML.

2) ***Yule-Walker for* ARMA(p, q)**. Method of moments. Not efficient.

3) **OLS for AR(p)**.

4) ***Innovations algorithm for* MA(q)**.

5) ***Hannan-Rissanen algorithm for* ARMA(p, q)**.

ARIMA Process – MLE

1) ***Maximun Likelihood Esimation* (MLE)**.

Steps:

1) Assume a distribution for the errors. Typically, *i.i.d.* normal, say:

$$\varepsilon_t \sim iid\ N(0, \sigma^2)$$

2) Write down the joint pdf for ε_t: $\quad f(\varepsilon_1, \ldots, \varepsilon_T) = f(\varepsilon_1) \ldots f(\varepsilon_T)$

<u>Note</u>: we are not writing the joint pdf in terms of the y_t's, as a multiplication of the marginal pdfs because of the dependency in y_t.

3) Get ε_t. For the general stationary ARMA(p, q) model:

$$\varepsilon_t = y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p} + \theta_1\ \varepsilon_{t-1} + \theta_2\ \varepsilon_{t-2} + \ldots + \theta_q\ \varepsilon_{t-q}$$

(if $\mu \neq 0$, demean y_t.)

4) The joint pdf for $\{\varepsilon_1, \ldots, \varepsilon_T)$ is:

$$f\left(\varepsilon_1, \cdots, \varepsilon_T \middle| \mu, \phi, \theta, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-T/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=1}^{n}\varepsilon_t^2\right\}$$

ARIMA Process – MLE

• Steps:

5) Let $Y = \{y_t\}$ and assume that initial conditions $Y_* = (y_{1-p}, \ldots, y_0)'$ and $\varepsilon_* = (\varepsilon_{1-q}, \ldots, \varepsilon_0)'$ are known.

6) The conditional log-likelihood function is given by

$$\ln L(\mu, \phi, \theta, \sigma^2) = -\frac{T}{2}\ln(2\pi\sigma^2) - \frac{S_*(\mu, \phi, \theta)}{2\sigma^2}$$

where $S_*(\mu, \phi, \theta) = \sum_{t=1}^{n} \varepsilon_t^2(\mu, \phi, \theta | Y, Y_*, \varepsilon_*)$ is the conditional SS.

Note: Usual Initial conditions: $y_* = \bar{y}$ and $\varepsilon_* = E[\varepsilon_t] = 0$.

• Numerical optimization problem. Initial values (y_*) matter.

ARIMA Process – MLE: AR(1)

Example:

- To change the joint from ε_t to y_t, we need the Jacobian, $|J|$:

$$|J| = \begin{vmatrix} \frac{\partial \varepsilon_2}{\partial Y_2} & \frac{\partial \varepsilon_2}{\partial Y_3} & \cdots & \frac{\partial \varepsilon_2}{\partial Y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon_n}{\partial Y_2} & \frac{\partial \varepsilon_n}{\partial Y_3} & \cdots & \frac{\partial \varepsilon_n}{\partial Y_n} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ -\phi & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix} = 1$$

$$f(y_1, y_2, \ldots, y_T) = f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T) * |J| = f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T)$$

- Then, the likelihood function can be written as

$$L(\phi, \sigma_a^2) = f(Y_1, \cdots, Y_n) = f(Y_1) f(Y_2, \cdots, Y_n | Y_1) = f(Y_1) f(\varepsilon_2, \cdots, \varepsilon_n)$$

$$= \left(\frac{1}{2\pi\gamma_0}\right)^{1/2} e^{-\frac{(Y_1-0)^2}{2\gamma_0}} \left(\frac{1}{2\pi\sigma^2}\right)^{(T-1)/2} e^{-\frac{1}{2\sigma^2}\sum_{t=2}^{T}(Y_t - \phi Y_{t-1})^2}, \quad \text{where } Y_1 \sim N\left(0, \gamma_0 = \frac{\sigma^2}{1-\phi^2}\right).$$

ARIMA Process – MLE: AR(1)

Example:

- Then,

$$L(\phi,\sigma^2)=\frac{\sqrt{1-\phi^2}}{(2\pi\sigma^2)^{n/2}}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{t=2}^{n}(Y_t-\phi Y_{t-1})^2+(1-\phi^2)Y_1^2\right]\right\}$$

- Then, the log likelihood function:

$$\ln L(\phi,\sigma^2)=-\frac{n}{2}\ln 2\pi-\frac{n}{2}\ln\sigma^2-\frac{1}{2}\ln(1-\phi^2)-$$

$$-\frac{1}{2\sigma^2}\underbrace{\left[\underbrace{\sum_{t=2}^{n}(Y_t-\phi Y_{t-1})^2}_{S_*(\phi)}+(1-\phi^2)Y_1^2\right]}_{S(\phi)}$$

- $S^*(\phi)$ is the conditional SS and $S(\phi)$ is the unconditional SS.

ARIMA Process – MLE: AR(1)

Example:

- F.o.c.'s:

$$\frac{\partial \ln L(\phi,\sigma^2)}{\partial\phi}=0$$

$$\frac{\partial \ln L(\phi,\sigma^2)}{\partial\sigma}=0$$

Note: If we neglect $ln(1-\phi^2)$, then MLE = Conditional LSE.

$$\max_{\phi} L(\phi,\sigma^2)=\min S(\phi).$$

If we neglect both $ln(1-\phi^2)$ and $(1-\phi^2)Y_1^2$, then

$$\max_{\phi} L(\phi,\sigma^2)=\min S(\phi_*).$$

ARIMA Process – Yule-Walker

2) ***Yule-Walker for*** **ARMA(p, q)**. Method of moments. Not efficient

• For an AR(p), the Yule-Walker estimator for $\boldsymbol{\phi}$ is given by solving

$$\boldsymbol{\gamma} = \boldsymbol{\Gamma}\,\boldsymbol{\phi}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(0) \end{bmatrix} \quad \text{a } p\text{x}p \text{ matrix}$$

$\boldsymbol{\phi}$ is the px1 vector of AR(p) coefficients
$\boldsymbol{\gamma}$ is the px1 vector of $\gamma(k)$ autocovariances.

• MM: Compute sample $\gamma(k)$'s –i.e., $\hat{\gamma}(k)$'s – & solve: $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\Gamma}}_p^{-1}\,\hat{\boldsymbol{\gamma}}_p$

• If $\hat{\gamma}(0) > 0$, then $\hat{\boldsymbol{\Gamma}}_p$ is non-singular.

ARIMA Process – Yule-Walker

• **Distribution:**

If y_t is an AR(p) process, and T is large,

$$\sqrt{T}\,(\hat{\phi} - \phi) \xrightarrow{a} \text{N}(0, \sigma^2\,\boldsymbol{\Gamma}^{-1})$$

• 100*(1 – α)% approximate C.I. for ϕ_j is $\hat{\phi}_j \pm z_{\alpha/2}\frac{\hat{\sigma}^2}{\sqrt{T}}(\hat{\boldsymbol{\Gamma}}_p{}^{-1})_{jj}^{1/2}$

Note: The Yule-Walker algorithm requires $\boldsymbol{\Gamma}^{-1}$.

• For AR(p). The **Levinson-Durbin (LD) algorithm** avoids $\boldsymbol{\Gamma}^{-1}$. It is a recursive linear algebra prediction algorithm. It takes advantage that $\boldsymbol{\Gamma}$ is a symmetric matrix, with a constant diagonal (Toeplitx matrix). Use LD replacing $\boldsymbol{\gamma}$ with $\hat{\boldsymbol{\gamma}}_p$.

• Side effect of LD: automatic calculation of PACF and MSPE.

ARIMA Process – Yule-Walker: AR(1)

Example: AR(1) (MM) estimation ($\mu = 0$):

$$y_t = \phi_1 y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim WN.$$

It is known that $\rho_1 = \phi$. Then, the MME of ϕ is

$$\Rightarrow \rho_1 = \hat{\rho}_1 .$$

$$\hat{\phi}_1 = \hat{\rho}_1 = \frac{\sum(Y_t-\bar{Y})(Y_{t+k}-\bar{Y})}{\sum(Y_t-\bar{Y})^2}$$

• Also, σ^2 is unknown:

$$\gamma(0) = \frac{\sigma^2}{(1-\phi_1^2)} \Rightarrow \hat{\sigma}^2 = \hat{\gamma}(0) * (1 - \hat{\phi}_1^2)$$

ARIMA Process – Yule-Walker: MA(1)

Example: MA(1) (MM) estimation: $y_t = \varepsilon_t - \theta\varepsilon_{t-1}$

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Again using the autocorrelation of the series at lag 1,

$$\rho_1 = -\frac{\theta}{(1+\theta^2)} = \hat{\rho}_1$$

$$\theta^2 \hat{\rho}_1 + \theta + \hat{\rho}_1 = 0$$

$$\hat{\theta}_{1,2} = \frac{-1 \pm \sqrt{1-4\hat{\rho}_1^2}}{2\hat{\rho}_1}$$

• Choose the root satisfying the invertibility condition. For real roots:

$$1 - 4\,\hat{\rho}_1{}^2 \geq 0 \Rightarrow 0.25 \geq \hat{\rho}_1{}^2 \Rightarrow -0.5 \leq \hat{\rho}_1{}^2 \leq 0.5$$

If $\hat{\rho}_1 = \pm\, 0.5$, unique real roots but non-invertible.
If $|\hat{\rho}_1| < 0.5$, unique real roots and invertible.

ARIMA Process – Yule-Walker

• Remarks
- The MMEs for MA and ARMA models are complicated.

- In general, regardless of AR, MA or ARMA models, the MMEs are sensitive to rounding errors. They are usually used to provide initial estimates needed for a more efficient nonlinear estimation method.

- The moment estimators are not recommended for final estimation results and should not be used if the process is close to being nonstationary or noninvertible.

ARIMA Process – Estimation Hannan-Rissanen

5) ***Hannan-Rissanen algorithm for* ARMA(p, q)**

Steps:
1. Estimate high-order AR.
2. Use Step (1) to estimate (unobserved) noise ε_t
3. Regress y_t against y_{t-1}, y_{t-2}, ..., y_{t-p}, $\hat{\varepsilon}_{t-1}$, ... , $\hat{\varepsilon}_{t-q}$
4. Get new estimates of ε_t. Repeat Step (3).

ARFIMA Process: Fractional Integration

- Consider a simple ARIMA model: $(1-L)^d\ y_t = \ \varepsilon_t$

- We went over two cases for $d = 0$ & 1. Granger and Joyeaux (1980) consider the model where $0 \leq d \leq 1$.

- We Taylor expand $(1-L)^d$ around $L_0 = 0$ (a binomial series expansion:

$$(1-L)^d = 1 - dL + \frac{d(d-1)L^2}{2!} - \frac{d(d-1)(d-2)L^3}{3!} + \ldots$$

- Similarly, for $(1-L)^{-d}$

$$(1-L)^{-d} = 1 + dL + \frac{(d+1)dL^2}{2!} + \frac{d(d+1)(d+2)L^3}{3!} \ldots$$

- Thus, the ARIMA(0, d, 1):

$$y_t = (1-L)^{-d}\varepsilon_t = \varepsilon_t + d\varepsilon_{t-1} + \frac{(d+1)d}{2!}\ \varepsilon_{t-2} + \ldots$$

ARFIMA Process: Fractional Integration

- In the ARIMA(0, d, 1):

$$y_t = (1-L)^{-d}\varepsilon_t = \varepsilon_t + d\varepsilon_{t-1} + \frac{(d+1)d}{2!}\ \varepsilon_{t-2} + \ldots$$

- The above MA can be approximated by:

$$y_t = (1-L)^{-d}\varepsilon_t \approx \varepsilon_t + (1+1)^d\varepsilon_{t-1} + (1+2)^{d-1}\ \varepsilon_{t-2} + \ldots$$
$$= \sum_{j=0}^{\infty} \beta_j \varepsilon_{t-j}$$

where $\beta_0 = 1$. Convergence depends on d.

- Hosking (1981) shows that $\rho(k) \propto k^{2d-1}$ as $k \to \infty$.

- The series is covariance stationary if $d < 1/2$. (When $d \geq 1/2$, the variance explodes.) Then, ARFIMA models have slow (hyperbolic to zero) decay patterns in the ACF.

ARFIMA Process: Fractional Integration

- $\rho(k) \propto k^{2d-1}$ as $k \to \infty$.

This type of slow decay patterns also show **long memory** for shocks. This type of process is neither $I(0)$ (stationary) nor $I(1)$ (unit root). It is an $I(d)$ (in between, no "short" memory, with decaying impact of shock, nor "persistent" memory, with permanent effect of shocks)!

• When $0 < d < 0.5$, the ARFIMA process is said to exhibit long memory, or **long-range positive dependence**. When $0 > d > -0.5$, the ARFIMA process is said to exhibit long **long-range negative dependence** (or **anti-persistence**).

Note: When $d = 0$, we have a stationary ARMA.

ARFIMA Process: Estimation

• Estimation is complicated. Many methods have been proposed. The majority of them are two-steps procedures. First, we estimate *d*. Then, we fit a traditional ARMA process to the transformed.

Popular estimation methods:
- Based on the log periodogram regressions, due to Geweke and Porter-Hudak (1983), GPH. Phillips (1999) has a generalized version of the GPH
- Rescaled range (RR), due to Hurst (1951) and modified by Lo (1991).
- Approximated ML (AML), due to Beran (1995). In this case, all parameters are estimated simultaneously.

ARFIMA Process: Remarks

• In a general review paper, Granger (1999) concludes that *ARFIMA* processes may fall into the **empty box** category –i.e., models with stochastic properties that do not mimic the properties of the data.

• Leybourne, Harris, and McCabe (2003) find some forecasting power for long series. Bhardwaj and Swanson (2004) find ARFIMA useful at longer forecast horizons.

ARFIMA Process: Example

• From Bhardwaj and Swanson (2004)

Table 2: Analysis of U.S. S&P500 Daily Absolute Returns (*)

Estimation Scheme and Forecast Horizon	ARFIMA Model	d	non-ARFIMA Model	DM	ENC-t	DM Best vs. RW
1 day ahead, recursive	WHI (1,1)	0.41 (0.0001)	ARMA(4,2)	-1.18	0.47	-13.64
5 day ahead, recursive	GPH (1,2)	0.57 (0.0011)	ARMA(4,2)	-0.71	1.75	-10.10
20 day ahead, recursive	GPH (1,2)	0.57 (0.0011)	ARMA(4,2)	-0.68	2.91	-5.96
120 day ahead, recursive	GPH (1,2)	0.57 (0.0011)	ARMA(4,2)	0.38	7.52	-6.33
240 day ahead, recursive	GPH (1,2)	0.57 (0.0011)	ARMA(4,2)	0.52	10.22	-6.16
1 day ahead, rolling	RR (1,1)	0.25 (0.0009)	ARMA(4,2)	2.02	4.56	-12.44
5 day ahead, rolling	GPH (1,2)	0.55 (0.0044)	ARMA(4,2)	-2.28	0.26	-10.24
20 day ahead, rolling	GPH (1,2)	0.55 (0.0044)	ARMA(4,2)	-2.44	0.79	-5.91
120 day ahead, rolling	GPH (1,2)	0.55 (0.0044)	ARMA(4,2)	-4.07	0.09	-6.32
240 day ahead, rolling	RR (1,1)	0.25 (0.0009)	ARMA(4,2)	-2.62	2.72	-5.90

Review: ARIMA Models – Box-Jenkins

• How do we select p, q, and d for an ARIMA model?

• Box-Jenkins Approach

1) Make sure data is stationary –check a time plot. If not, differentiate.

2) Use IC (AIC, AICc, BIC, etc.). You can also use ACF & PACF to guess small values for p & q. If ACF shows seasonal patterns (waves or periodic significant pikes), remove them.

3) Estimate order p, q. (ML, Hannan-Rissanen algorithm, etc.)

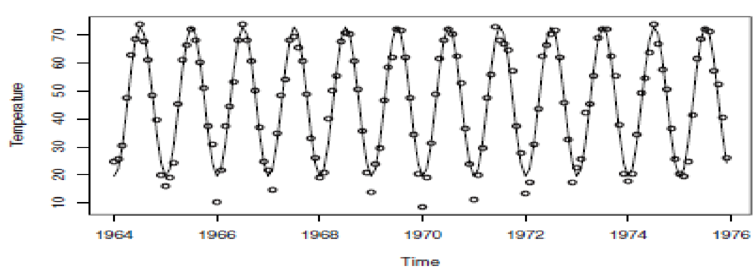
4) Run diagnostic tests on residuals (Check ACF, LB tests).
$\Rightarrow$ Are they white noise? If not, add lags (p or q, or both).

• Value parsimony. When in doubt, keep it simple (KISS).

• Looks simple, but there are a lot of nuances to the process. Step 1 is crucial: Need to remove everything deterministic.

Seasonal Patterns

• We say a time series shows seasonal patterns if it repeats itself after a regular period of time.



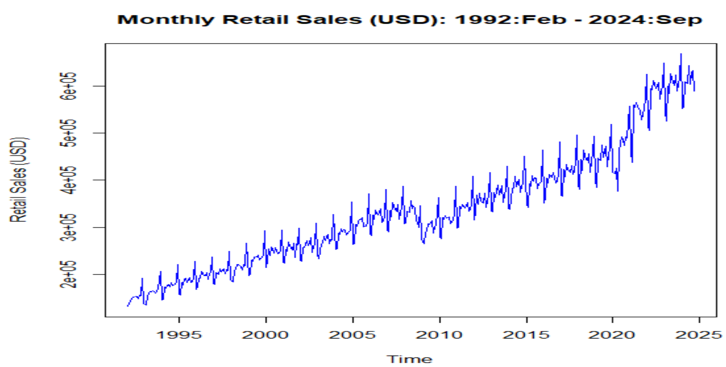
• "Business cycle effects" in macroeconomics "time of the day" in trading patterns, "Monday effect" for stock returns, "9 to 5 effect" for electricity demand, etc.

• The smallest time period for this repetitive phenomenon is called a seasonal period, s.

Seasonal Patterns

- In the Box-Jenkins approach, we also incorporate seasonal patterns.

- Some seasonal patterns are very clear. For example, retail sales increase in December, hotel occupancy goes up in the Summer, etc.



Seasonal Patterns: Additive & Multiplicative

- In time series, seasonal patterns ("**seasonalities**") can show up in two forms: additive and multiplicative.

(1) Additive: The seasonal variation is independent of the level. The amplitude of the seasonal pattern is constant over time. The constant amplitude can be around a mean or constant around a trend.

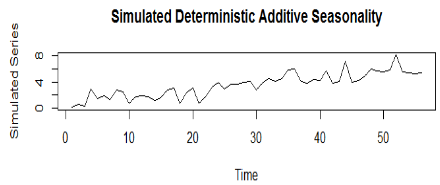
(2) Multiplicative: The seasonal variation is a function of the level. Thus, we see an increasing amplitude in the seasonal variation over time. Again, the increasing amplitude can be around a mean or around a trend.

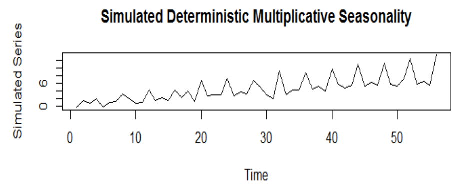
Note: In practice, because of the presence of the error term, we expect to see the constant or increasing amplitude on average.

Seasonal Patterns: Additive & Multiplicative

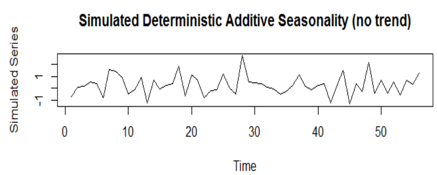
Examples: We simulate the two seasonal patterns, additive and multiplicative, with trend and no trend.

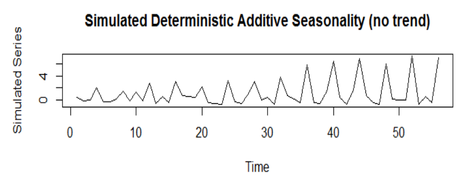
A. With trend





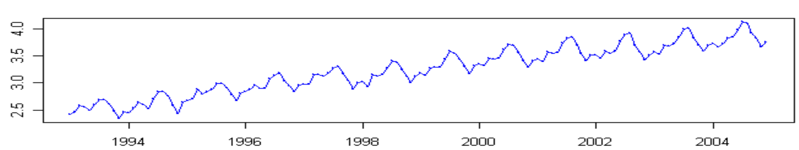
B. With no trend

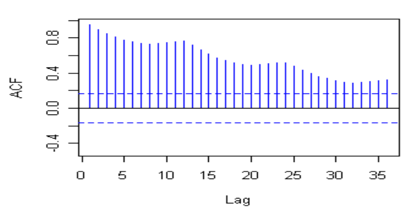


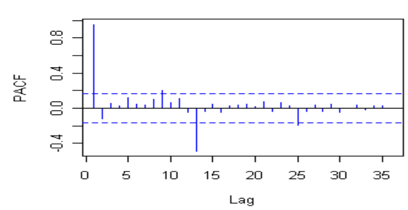


Discovering Seasonal Patterns

• In general, seasonal patterns are evident in a plot of the raw series. Also, the ACF and PACF can be used to discover the pattern.







Signs: Periodic repetitive wave pattern in ACF, repetition of significant ACFs, PACFs after s periods.

Removing Seasonal Patterns

• In the presence of seasonal patterns, we proceed to do seasonal adjustments to remove these predictable influences.

• Seasonalities can blur both the true underlying movement in the series, as well as certain non-seasonal characteristics which may be of interest to analysts.

• Similar to the trend, the type of adjustment depends on how we view the seasonal pattern: **Deterministic** or **Stochastic**.

- **Deterministic** – Usual treatment: Build a deterministic function:

$$f(s) = f(t + k * s), \qquad k = 0, \pm 1, \pm 2, \ldots$$

- **Stochastic** – Usual treatment: SARIMA model. For example:

$$(1 - L^s)\phi(L)(1 - L)^d \; y_t = \theta(L)\Theta(L^s)\varepsilon_t$$

where s the seasonal periodicity or frequency of y_t.

Removing Seasonalities: Deterministic Case

• We follow a similar 2-step process to detrending:
1) Regress y_t against the seasonal dummies. Keep residuals
2) With the residuals, follow Box-Jenkins to select an ARIMA model.

• For **Step 1**. Suppose y_t has monthly frequency, we suspect that y_t increases every December.
– For the **additive model**, we regress y_t against a constant and a December dummy, $\boldsymbol{D}_t$:

$$y_t = \mu + \boldsymbol{D}_t \; \boldsymbol{\mu}_s + \varepsilon_t$$

– For the **multiplicative model**, we regress y_t against a constant and a December dummy, $\boldsymbol{D}_t$, interacting with a trend:

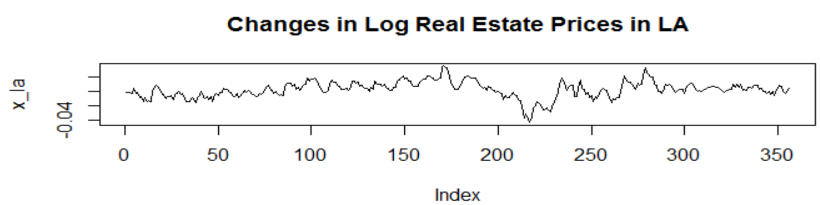
$$y_t = \mu + \boldsymbol{D}_t \; \boldsymbol{\mu}_s * t + \varepsilon_t$$

• For **Step 2**. Use the residuals of these regressions, e_t, –i.e., e_t = *filtered* y_t, free of "monthly seasonal effects"– for ARMA modeling.

Removing Seasonalities: Deterministic Case

Example: We model **log changes in real estate prices in the LA market**, y_t. First, we run a regression to remove (filter) the monthly effects from y_t. Then, we model y_t as an ARMA(p, q) process.

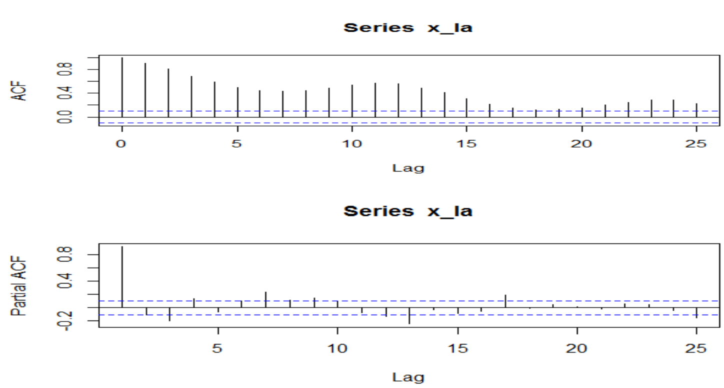
```
RE_da <- read.csv(" http://www.bauer.uh.edu/rsusmel/4397/Real_Estate_2019.csv",
head=TRUE, sep=",")
x_la <- RE_da$LA_c
zz <- x_la
T <- length(zz)
plot(x_la, type="l", main="Changes in Log Real Estate Prices in LA")
```



Removing Seasonalities: Deterministic Case

Example (continuation): We look at the ACF & PACF for LA

```
> acf(x_la)
> pacf(x_la)
```



Note: ACF shows highly autocorrelated data, with some seasonal pattern (there is a periodic decreasing wave).

Removing Seasonalities: Deterministic Case

Example (continuation): We define monthly dummies

```
Feb1 <- rep(c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create January dummy
Mar1 <- rep(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create March dummy
Apr1 <- rep(c(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create April dummy
May1 <- rep(c(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create May dummy
Jun1 <- rep(c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create June dummy
Jul1 <- rep(c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Jul dummy
Aug1 <- rep(c(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Aug dummy
Sep1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0), (length(zz)/12+1))     # Create Sep dummy
Oct1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Nov1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Dec1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0), (length(zz)/12+1))     # Create Oct dummy
seas1 <- cbind(Feb1, Mar1, Apr1, May1, Jun1, Jul1, Aug1, Sep1, Oct1, Nov1, Dec1)
seas <- seas1[1:T,]

x_la_fit_sea <- lm(x_la ~ seas)          # Regress x_la against constant + seasonal dummies
> summary(x_la_fit_sea)
```

Removing Seasonalities: Deterministic Case

Example (continuation): We define monthly dummies

```
> summary(x_la_fit_sea)
Coefficients:
              Estimate   Std. Error t value Pr(>|t|)
(Intercept) -0.0014063  0.0020125  -0.699 0.485157
seasFeb1     0.0006752  0.0028223   0.239 0.811079
seasMar1     0.0049095  0.0028223   1.740 0.082838 .
seasApr1     0.0090903  0.0028223   3.221 0.001400 **
seasMay1     0.0104159  0.0028223   3.691 0.000260 ***
seasJun1     0.0103464  0.0028223   3.666 0.000285 ***
seasJul1     0.0080593  0.0028223   2.856 0.004557 **
seasAug1     0.0062247  0.0028223   2.206 0.028080 *
seasSep1     0.0032244  0.0028223   1.142 0.254055
seasOct1     0.0011967  0.0028461   0.420 0.674421
seasNov1    -0.0006218  0.0028461  -0.218 0.827181
seasDec1    -0.0009031  0.0028461  -0.317 0.751195
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: Returns –i.e., home prices– are higher from April to August.

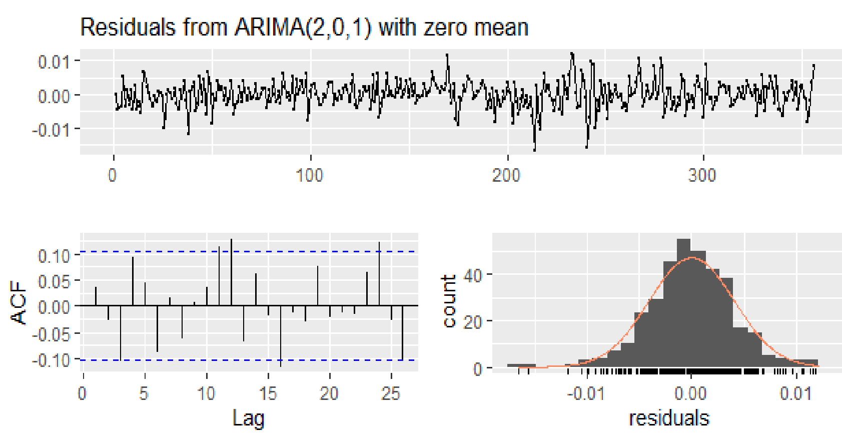
Removing Seasonalities: Deterministic Case

Example (continuation): Now, we model e_t, the filtered LA series

```
x_la_filt <- x_la_fit_sea$residuals                 # residuals, e_t = filtered x_la series
fit_ar_la_filt <- auto.arima(x_la_filt)             # use auto.arima to look for a good model
> fit_ar_la_filt
Series: x_la_filt
ARIMA(2,0,1) with zero mean

Coefficients:
         ar1     ar2     ma1
      0.0987  0.7737  0.7245
s.e.  0.0963  0.0866  0.1136

sigma^2 estimated as 1.668e-05:  log likelihood=1453.66
AIC=-2899.33   AICc=-2899.21   BIC=-2883.83

> checkresiduals(fit_ar_la_filt)

        Ljung-Box test

data:  Residuals from ARIMA(2,0,1) with zero mean
Q* = 13.5, df = 7, p-value = 0.06083       ⇒ Reject H0 at 5% lever. But, judgement call is OK.

Model df: 3.   Total lags used: 10
```

Removing Seasonalities: Deterministic Case

Example (continuation): We check residual plots.



Note: ACF shows some small, but significant autocorrelations, but the seasonal (wave) pattern is no longer there.

Removing Seasonalities: SARIMA

• For stochastic seasonality, we use the Seasonal ARIMA model. In general, we have the SARIMA$(P,D,Q)_s$:

$$\Phi_P(L^s)\,(1-L^s)^D\; y_t = \theta_0 + \Theta_Q(L^s)\varepsilon_t$$

where θ_0 is constant and

$$\Phi_P(L^s) = 1 - \Phi_1 L^s - \Phi_2 L^{2s} - \Phi_3 L^{3s} \ldots - \Phi_P L^{Ps}$$

$$\Theta_Q(L^s) = 1 - \Theta_1 L^s - \Theta_2 L^{2s} - \Theta_3 L^{3s} \ldots - \Theta_Q L^{Qs}$$

Example 1: SARIMA$(0,0,1)_{12}$ = SMA$(1)_{12}$

$$y_t = \theta_0 + \varepsilon_t - \Theta\,\varepsilon_{t-12}$$

- Invertibility Condition: $|\Theta| < 1$.

– $E[y_t] = \theta_0$.

– $Var[y_t] = (1+\Theta^2)\sigma^2$

$$ACF: \rho_k = \begin{cases} \dfrac{-\Theta}{1+\Theta^2}, & |k| = 12 \\ 0, & \text{otherwise} \end{cases}$$

Removing Seasonalities: SARIMA

Example 2: SARIMA$(1,0,0)_{12}$= AR$(1)_{12}$

$$(1-\Phi L^{12})\; y_t = \theta_0 + \varepsilon_t$$

- This is a simple seasonal AR model.

- Stationarity Condition: $|\Phi| < 1$.

– $E[y_t] = \frac{\theta_0}{1-\Phi}$

– $Var[y_t] = \frac{\sigma^2}{1-\Phi^2}$

– ACF: $\rho(12*k) = \Phi^{12*k}$ $\qquad k = 0, \pm1, \pm2, \ldots$

• When $\Phi = 1$, the series is non-stationary. To test for a unit root, consider seasonal unit root tests.

Seasonal Time Series – Multiplicative SARIMA

• A special, parsimonious class of seasonal time series models that is commonly used in practice is the multiplicative seasonal model $ARIMA(p,\, d,\, q)(P,D,Q)_s$:

$$\phi_p(L)\Phi_P(L^s)(1-L)^d(1-L^s)^D\; y_t = \theta_0 + \theta_q(L)\Theta_Q(L^s)\varepsilon_t$$

where all zeros of $\phi_p(L)$; $\Phi_P(L^s)$; $\theta_q(L)$ & $\Theta_Q(L^s)$ lie outside the unit circle. Of course, there are no common factors between $\phi_p(L)\Phi_P(L^s)$ and $\theta_q(L)\Theta_Q(L^s)$

• When $\Phi_P(L^s=1)=0$, the series is non-stationary. To test for a unit root, consider seasonal unit root tests.

Seasonal Time Series – Multiplicative SARIMA

• We derive ACF as usual: For example,

$$W_t = (1-\theta L)\,(1-\Theta L^{12})\,y_t, \qquad W_t \sim I(0)$$

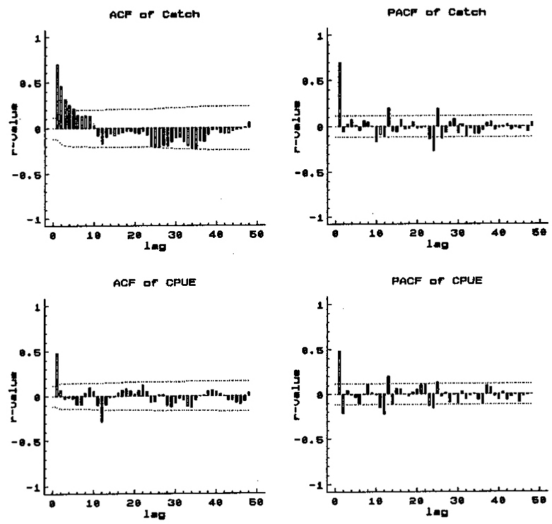
then,

$$\begin{aligned} W_t &= (1-\theta L)\,(1-\Theta L^{12})\,y_t \\ &= (1-\theta L-\Theta L^{12}+\theta\Theta L^{13}) \\ &= y_t-\theta\, y_{t-1}-\Theta\, y_{t-12}+\theta\Theta\, y_{t-13} \end{aligned}$$

$$\gamma_k = \begin{cases} (1+\theta^2)(1+\Theta^2)\sigma^2, & k=0 \\ -\theta(1+\Theta^2)\sigma^2, & |k|=1 \\ -\Theta(1+\theta^2)\sigma^2, & |k|=12 \\ \theta\Theta\sigma^2, & |k|=11,13 \\ 0, & otherwise \end{cases} \qquad \rho_k = \begin{cases} \dfrac{-\theta}{(1+\theta^2)}, & |k|=1 \\ \dfrac{-\Theta}{(1+\Theta^2)}, & |k|=12 \\ \dfrac{\theta\Theta}{(1+\theta^2)(1+\Theta^2)}, & |k|=11,13 \\ 0, & otherwise \end{cases}.$$

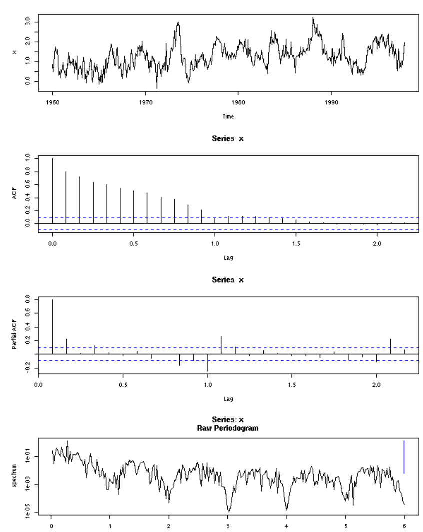
Seasonal Time Series – Multiplicative SARIMA

• The ACF and PACF can be used to discover seasonal patterns.



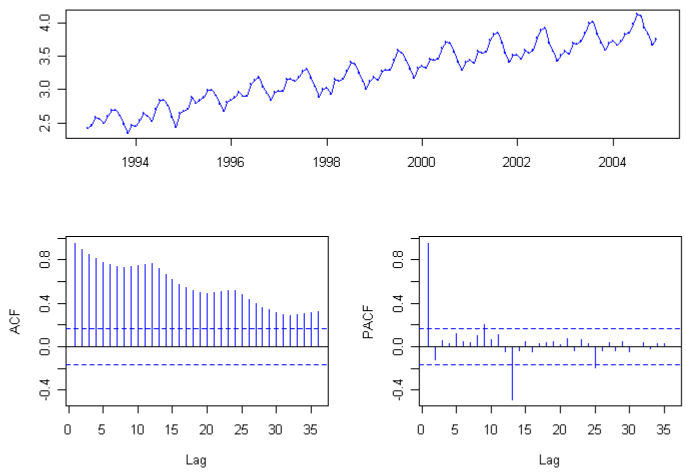
Seasonal Time Series – Multiplicative SARIMA

• The ACF and PACF can be used to discover seasonal patterns.



Seasonal Time Series – Multiplicative SARIMA

• The ACF and PACF can be used to discover seasonal patterns.



Seasonal Time Series – Multiplicative SARIMA

• Most used seasonal model in practice: SARIMA(0,1,1)(0,1,1)$_{12}$

$$(1-L)\,(1-L^{12})\;y_t = (1-\theta L)\,(1-\Theta L^{12})\;\varepsilon_t$$

where $|\theta| < 1$ and $|\Theta| < 1$.

• This model is the most used seasonal model in practice. It was used by Box and Jenkins (1976) for modeling the well-known monthly series of airline passengers. It is called the *airline model*.

• We usually work with the RHS variable,

$$W_t = (1-L)\,(1-L^{12})\;y_t$$

$(1-L)$: "regular" difference

$(1-L^{12})$: "seasonal" difference.

Seasonal Time Series – Seasonal Unit Roots

• If a series has seasonal unit roots, then standard ADF test statistic do not have the same distribution as for non-seasonal series.

• Furthermore, seasonally adjusting series which contain seasonal unit roots can alias the seasonal roots to the zero frequency, so there is a number of reasons why economists are interested in seasonal unit roots.

• See Hylleberg, S., Engle, R.F., Granger, C. W. J., and Yoo, B. S., Seasonal integration and cointegration,(1990, *Journal of Econometrics*).

Non-Stationarity in Variance

• Stationarity in mean does not imply stationarity in variance

• Non-stationarity in mean implies non-stationarity in variance.

• If the mean function is time dependent:
1. The variance, $Var[y_t]$, is time dependent.
2. $Var[y_t]$ is unbounded as $t \to \infty$.
3. Autocovariance functions and ACFs are also time dependent.
4. If t is large with respect to y_0, then $\rho_k \approx 1$.

• It is common to use **variance stabilizing transformations**: Find a function $G(.)$ so that the transformed series $G(y_t)$ has a constant (or lower) variance. For example, the Box-Cox transformation:

$$G(y_t) = \frac{({y_t}^{\lambda} - 1)}{\lambda}$$

Non-Stationarity in Variance

• Many times, this stabilizing transformation is done because the variance is non-stationary. In practice, a variance stabilizing transformation is done to reduce the variance of the series.

• Traditionally, variance stabilizing transformations are used when working with a nominal series (not changes, say, USD total retail sales or total units sold).

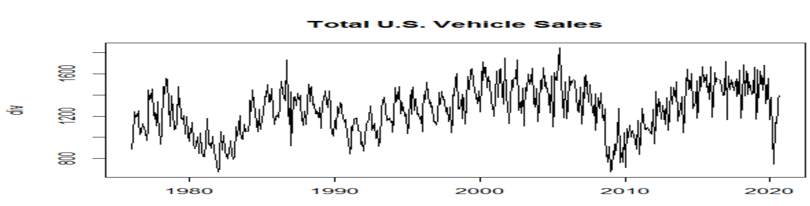
• In the context of nominal series, the most popular transformation is the log:

$$G(y_t) = \log(y_t)$$

Non-Stationarity in Variance: Logs

Example: We log transform the monthly variable Total U.S. Vehicle Sales data (1976: Jan – 2020: Sep):

```
Car_da <- read.csv("https://www.bauer.uh.edu/rsusmel/4397/TOTALNSA.csv", head=TRUE, sep=",")
x_car <- Car_da$TOTALNSA

library(tseries)
ts_car <- ts(x_car,start=c(1976,1),frequency=12)
plot.ts(ts_car,xlab="Time",ylab="div", main="Total U.S. Vehicle Sales")
```

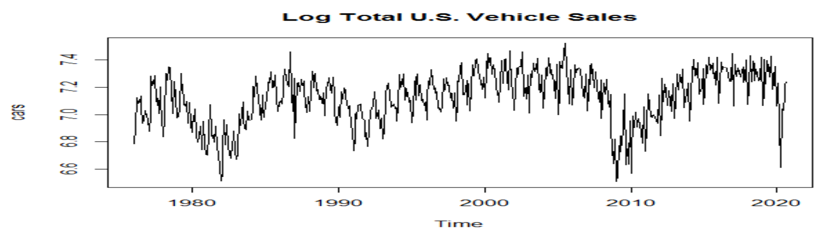



```
> mean(x_car)
[1] 1260.818
> sd(x_car)
[1] 225.5706
```

Non-Stationarity in Variance: Logs

Example (continuation):

```
l_car <- log(ts_car)
> plot.ts(l_car,xlab="Time",ylab="div", main="Log Total U.S. Vehicle Sales")
```




```
> mean(l_car)
[1] 7.122416
> sd(l_car)
[1] 0.1889378
```

Note: Big reduction in volatility. Though pattern of series not significantly changed.

Variance Stabilizing Transformation - Remarks

• Variance stabilizing transformation is only for positive series. If a series has negative values, then we need to add each value with a positive number so that all the values in the series are positive.

• Then, we can search for any need for transformation.

• It should be performed before any other analysis, such as differencing.

• Not only stabilize the variance, but we tend to find that it also improves the approximation of the distribution by Normal distribution.