Forecasting
Part 2

JY Le Boudec
6. Differencing Filters
7. Filters for dummies
8. Prediction with filters
9. ARMA Models
10. Other methods
6. Differencing the Data

- We have seen that changing the scale of the data may be important for obtaining a good model.

- Another kind of pre-processing is the application of filters. The idea is that it may be simpler to forecast the filtered data.

- A filter (in full, discrete-time causal filter) is a mapping from the set of finite-length time series to the same set. By convention, we consider that a filter keeps the length of the time-series unchanged. Further, a filter has to be linear, time-invariant and causal. The latter means that output of the filter up to time $t$ depends only on the input up to time $t$. 
Differencing filter $\Delta_1$

- Differencing filter $\Delta_1 = \text{discrete-time derivative}$
  
  $$Y = (Y_1, \ldots, Y_t) \mapsto \Delta_1 Y = (Y_1, Y_2 - Y_1, \ldots, Y_t - Y_{t-1})$$

  $$\Delta_1 Y = X \iff X \text{ has same length as } Y \text{ and } X_t = Y_t - Y_{t-1}$$

  with the convention that $Y_t = 0$ whenever $t \leq 0$

  $\Delta_1$ is a filter, thus is linear, $\Delta_1 (Y + Z) = \Delta_1 Y + \Delta_1 Z$

  If $Y_t = Z_t + \alpha t$ then $(\Delta_1 Y)_t = (\Delta_1 Z)_t + \alpha$: $\Delta_1$ removes linear trends

  Repeated application of $\Delta_1$ removes polynomial trends
De-seasonalizing filters

- De-seasonalizing $R_s$: (sum of last $s$ values)
  \[ R_s Y = X \iff X \text{ has same length as } Y \text{ and } X_t = Y_{t-s+1} + \cdots + Y_{t-1} + Y_t \]
  with the convention that $Y_t = 0$ whenever $t \leq 0$

If $Y_t$ is periodic of period $s$ then $R_s Y$ is constant

$R_s$ removes periodic components

- Differencing $\Delta_s$:
  \[ \Delta_s Y = X \iff X \text{ has same length as } Y \text{ and } X_t = Y_t - Y_{t-s} \]
  with the convention that $Y_t = 0$ whenever $t \leq 0$
De-seasonalizing filters

\[ \Delta_s = R_s \Delta_1 \]

this means that if \( Y \rightarrow Z \rightarrow X \) and \( Y \rightarrow X' \) then \( X = X' \)

Proof: 

\[ Z_t = Y_t - Y_{t-1} \]

\[ X_t = Z_t + \ldots + Z_{t-s+1} \]

\[ = Y_t - Y_{t-1} + Y_{t-1} - Y_{t-2} \ldots + Y_{t-s+1} - Y_{t-s} = Y_t - Y_{t-s} \]
Say what is true

A. A
B. B
C. C
D. A,B
E. A,C
F. B,C
G. All
H. None
I. I don’t know

\[ A. R_s \Delta_1 = \Delta_s \]
\[ B. \Delta_1 R_s = \Delta_s \]
\[ C. \Delta_1 \Delta_1 = \Delta_2 \]
Solution

A is true as we saw earlier

B is true. We prove by a direct computation as we did earlier, or what we can use the property that any two filters commute, i.e. the order in which a succession of filters is applied does not matter: \( FG = GF \) for any two filters \( F, G \)

C is false. Let us compute \( X = \Delta_1 \Delta_1 Y \). Let \( Z = \Delta_1 Y \), so that

\[
X_t = Z_t - Z_{t-1}
\]

\[
= (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}
\]

which is not equal to \( Y_t - Y_{t-2} \)

\( \Delta_1 \Delta_1 \), also noted \((\Delta_1)^2\) is the discrete-time second derivative.
Example 5.3: Internet Traffic. In Figure 5.7 we apply the differencing filter \( \Delta_1 \) to the time series in Example 5.1 and obtain a strong seasonal component with period \( s = 16 \). We then apply the de-seasonalizing filter \( R_{16} \); this is the same as applying \( \Delta_{16} \) to the original data. The result does not appear to be stationary; an additional application of \( \Delta_1 \) is thus performed.
Point Predictions from Differenced Data

How are these predictions made? To answer this question, we need to see how to use filters.
D.1.1 Backshift Operator

We consider data sequences of finite, but arbitrary length and call \( S \) the set of all such sequences (i.e. \( S = \bigcup_{n=1}^{\infty} \mathbb{R}^n \)). We denote with \( \text{length}(X) \) the number of elements in the sequence \( X \).

The backshift operator is the mapping \( B \) from \( S \) to itself defined by:

\[
\text{length}(BX) = \text{length}(X) \\
(BX)_1 = 0 \\
(BX)_t = X_{t-1} \quad t = 2, \ldots, \text{length}(X)
\]

We usually view a sequence \( X \in S \) as a column vector, so that we can write:

\[
B \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} 0 \\ X_1 \\ \cdots \\ X_{n-1} \end{pmatrix} \quad (D.1)
\]

when \( \text{length}(X) = n \).
 DEFINITION D.1. A filter (also called “causal filter”, or “realizable filter”) is any mapping, say $F$, from $S$ to itself that has the following properties.

1. A sequence of length $n$ is mapped to a sequence of same length.
2. There exists an infinite sequence of numbers $h_m$, $m = 0, 1, 2, ...$ (called the filter’s impulse response) such that for any $X \in S$

$$\begin{align*}
(FX)_t &= h_0X_t + h_1X_{t-1} + \cdots + h_{t-1}X_1 & t &= 1, \ldots, \text{length}(X) 
\end{align*}$$

(D.4)

In matrix form, if we know that $\text{length}(X) \leq n$ we can write Eq.(D.4) as

$$FX = \begin{pmatrix}
h_0 & 0 & \cdots & 0 & 0 \\
h_1 & h_0 & \cdots & \cdots & \cdots \\
h_2 & h_1 & \cdots \\
\vdots & \vdots & \cdots & h_0 & 0 \\
h_{n-1} & h_{n-2} & \cdots & h_1 & h_0
\end{pmatrix} X$$

(D.6)
<table>
<thead>
<tr>
<th>Notation for filter $F$</th>
<th>Formula for $Y = FX$</th>
<th>Impulse Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$Y_t = X_t$</td>
<td>$(1,0,0,...)$</td>
</tr>
<tr>
<td>$B$</td>
<td>$Y_t = X_{t-1}$</td>
<td>$(0,1,0,0,...)$</td>
</tr>
<tr>
<td>$\Delta_1 = 1 - B$</td>
<td>$Y_t = X_t - X_{t-1}$</td>
<td>$(1,-1,0,0,...)$</td>
</tr>
<tr>
<td>$\Delta_s = 1 - B^s$</td>
<td>$Y_t = X_t - X_{t-s}$</td>
<td>$(1,0,...,0,-1,0,0,...)$</td>
</tr>
<tr>
<td>$R_s = 1 + B + \cdots + B^{s-1}$</td>
<td>$Y_t = X_t + \cdots + X_{t-s+1}$</td>
<td>$(1,...,1,0,0,...)$</td>
</tr>
<tr>
<td>$\frac{1}{F}$</td>
<td>Inverse filter: $Y = \left(\frac{1}{F}\right)X$ means $X = FY$</td>
<td></td>
</tr>
</tbody>
</table>
\[ FX = \begin{pmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & \ddots & \vdots & \vdots \\ h_2 & h_1 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & h_0 & 0 \\ h_{n-1} & h_{n-2} & \cdots & h_1 & h_0 \end{pmatrix} X \]

**D.1.5 Inverse of Filter**

Since the matrix in Eq.(D.6) is triangular, it is invertible if and only if its diagonal terms are non-zero, i.e. if \( h_0 \neq 0 \), where \( h_0 \) is the first term of its impulse response. If this holds, it can also be seen that the reverse mapping \( F^{-1} \) is a filter, i.e. it satisfies the conditions in Definition (D.1). Thus a filter \( F \) is invertible if and only if \( h_0 \neq 0 \).
D.1.7 Calculus of Filters

When the filter $F'$ is invertible, the composition $F(F'^{-1})$ is also noted $\frac{F}{F'}$. There is no ambiguity since composition is commutative, namely

$$\frac{F}{F'} = F(F'^{-1}) = (F'^{-1})F$$  \hspace{1cm} (D.17)

We have thus defined the product and division of filters. It is straightforward to see that the addition and subtraction of filters are also filters. For example, the filter $F + F'$ has impulse response $h_m + h'_m$ and the filter $-F$ has impulse response $-h_m$. 
Let $F$ be the filter defined by $Y = FX$ with

$$Y_t = X_t - 3X_{t-1} + 2X_{t-2}$$

Say what is true

A. $F = 1 - 3B + 2B^2$

B. The impulse response of $F$ is
   \((1, -3, 2, 0, 0, ... )\)

C. A and B

D. None

E. I don’t know
Solution

Both A and B are true, by definition of filters
Let $F$ be the filter defined by $Y = FX$ with

$$Y_t - 3Y_{t-1} + 2Y_{t-2} = X_t$$

Say what is true

A. $F = \frac{1}{1-3B+2B^2}$

B. The impulse response of $F$ is $(1, -3, 2, 0, 0 ...)$

C. A and B

D. None

E. I don’t know
Solution

A is true. Indeed we can write the definition of $F$ as

$$(1 - 3B + 2B^2)Y = X$$

Now the filter $(1 - 3B + 2B)$ is invertible (the coefficient $h_0$ is non-zero) therefore $Y = \frac{1}{1-3B+2B^2} X$

B is false. What is given is the impulse response of the inverse filter.
\( Y = \text{filter}(P,Q,X) \) computes the output \( Y = [y_1 \ y_2 \ y_3 \ldots \ y_n] \) of the filter, where \( P = [P_0 \ P_1 \ P_2 \ldots P_p] \), \( Q = [1 \ Q_1 \ Q_2 \ldots Q_q] \) are the filter coefficients and \( X = [x_1 \ x_2 \ x_3 \ldots] \) is the input. The filter is defined by the relation

\[
y_k + Q_1 y_{k-1} + \ldots + Q_q y_{k-p} = P_0 x_k + P_1 x_{k-1} + \ldots + P_q x_{k-q}
\]

where we set \( x_i = 0 \) and \( y_i = 0 \) when \( i < 0 \) or \( i > n \).

The polynomial \( P(\xi) = P_0 \xi^p + P_1 \xi^{q-1} + \ldots + P_q \) is called the **numerator polynomial** and \( Q(\xi) = \xi^q + Q_1 \xi^{q-1} + \ldots + Q_q \) the **denominator polynomial**.

In our terminology, this filter is the mapping

\[
\mathbb{R}^n \rightarrow \mathbb{R}^n
\]

\[
X \rightarrow Y = \frac{\sum_{i=0}^{p} P_i B^i}{1 + \sum_{j=1}^{q} Q_j B} (X) = \frac{P(B)}{Q(B)} \cdot X
\]

<table>
<thead>
<tr>
<th>Matlab</th>
<th>Filter Notation</th>
<th>Equation</th>
</tr>
</thead>
</table>
| \( Y = \text{filter}([0.1 \ 0.2 \ 0.3], \ [1 \ -0.2], \ X) \) | \( Y = \frac{0.1+0.2B+0.3B^2}{1-0.2B} \cdot X \) | \( Y_t - 0.2Y_{t-1} \)
|  |  | \( = 0.1X_t + 0.2X_{t-1} \)
|  |  | \( + 0.3 \ X_{t-2} \) |
A sample of \[ Y = \frac{0.1 + 0.2B + 0.3B^2}{1 - 0.2B} X \]

Q: how can we compute \( X \) back from \( Y \)?

A: inverse the filter \( X = \frac{1 - 0.2B}{0.1 + 0.2B + 0.3B^2} Y \)

The inverse of \( Y = \text{filter}(P, Q, X) \) is \( X = \text{filter}(Q, P, X) \)
defined if first element of \( Q \) is \( \neq 0 \)

The result is shown with green dots; after \( t = 60 \) the results are incorrect. Why?
To understand what happens, let us compute the coefficients of these filters (i.e. their impulse responses)

It is obtained by $h = \text{filter}(P, Q, \text{imp})$ where $\text{imp} = [1 \ 0 \ 0 \ \ldots]$ is called an impulse

The impulse of $F^{-1}$ grows exponentially and becomes huge → numerical computation becomes impossible
Filter Stability: \( \sum_{n} |h_n| < \infty \)

- A filter that is unstable usually causes numerical problems (accumulation of rounding errors)

- A filter of this form is stable if and only if all its *poles* are inside the unit disk (their modulus is less than 1). The poles are the (usually complex) roots of the denominator polynomial \( Q(\xi) = \xi^q + Q_1\xi^{q-1} + \ldots + Q_q \). The *zeroes* of the filter are the roots of the numerator polynomial \( P(\xi) = P_0\xi^p + P_1\xi^{q-1} + \ldots + P_q \). They are the poles of the reverse filter. Thus, the reverse filter is stable if the zeroes of the original filter are all inside the unit disk.

*zplane* \((P, Q)\) plots the zeroes and the poles of the filter, together with the unit circle.

Zeroes of \( F \)

Solutions of \( 0.1z^2 + 0.2z + 0.3 = 0 \)

Pole of \( F \) = \( \frac{0.1 + 0.2B + 0.3B^2}{1-0.2B} \)

Solution of \( z - 0.2 = 0 \)
A filter with stable inverse

\[ P = [0.5 \ 0.3 \ 0.2] \quad Q = [1] \]
What is true about this filter $F$ (where $Y = FX$)?

- $P = [1]$ $Q = [0.5 \ 0.3 \ 0.2]$

A. $Y_t = X_t - 0.5Y_t - 0.3Y_{t-1} - 0.2Y_{t-2}$

B. $Y_t = \frac{X_t}{0.5Y_t + 0.3Y_{t-1} + 0.2Y_{t-2}}$

C. A and B

D. None

E. I don’t know
Solution

A is true

B is false. It is true that $Y_t = \left(\frac{1}{0.5 + 0.3B + 0.2 B^2} X\right)_t$

Answer A
MA and AR representation of a filter $F$

Let $Y = FX$

**Definition:** Moving Average representation $F$

$$Y_t = h_0 X_t + h_1 X_{t-1} + \cdots + h_{t-1} X_1$$

This is the standard representation and $h_i$ is the impulse response.
We say that $F$ is an MA($q$) filter if $h_i = 0$ for $i > q$

**Definition:** Auto-Regressive representation

$$Y_t = c_0 X_t - c_1 Y_{t-1} - c_2 Y_{t-2} - \cdots - c_{t-1} Y_1$$

It follows from the impulse response of $F^{-1}$:

$$X_t = h'_0 Y_t + h'_1 Y_{t-1} + \cdots + h'_{t-1} Y_1$$
Example: the filter $L = \Delta_1 \Delta_{16}$

$L = (1 - B)(1 - B^{16}) = 1 - B - B^{16} + B^{17}$

i.e.
$$Y_t = X_t - X_{t-1} - X_{t-16} + X_{t-17}$$

$L$ is a MA(17) filter

Let us compute the AR representation of $Y$; we obtain it from the impulse response of the reverse filter; let us solve for $X$ in $Y = LX$. After some math we find $h'_n = 1 + \left[\frac{n}{16}\right]$, i.e.

$$X_t = Y_t + Y_{t-1} + \cdots + Y_{t-15} + 2(Y_{t-16} + Y_{t-17} + \cdots + Y_{t-31}) + 3(Y_{t-32} + Y_{t-33} + \cdots + Y_{t-47}) + \cdots$$

i.e. $Y_t = X_t - Y_{t-1} - \cdots - Y_{t-15} - 2Y_{t-16} - \cdots - 3Y_{t-47} - \cdots$
8. How is this prediction done?

Recall that $X = LY$ with $L = \Delta_1 \Delta_16$ and we assume $X \sim \text{iid } F()$

thus

$X_t = Y_t - Y_{t-1} - Y_{t-16} + Y_{t-17}$

$Y_t = X_t + Y_{t-1} + Y_{t-16} - Y_{t-17}$

(MA representation of $L$, i.e. AR representation of $L^{-1}$)

Prediction at lag $\ell = 1$: assume we know $Y_1, \ldots, Y_t$

$Y_{t+1} = X_{t+1} + Y_t + Y_{t-15} - Y_{t-16}$

Given the past up to time $t$, this is random with distribution $F()$
Point Prediction at lag 1

- **Prediction at lag $\ell = 1$:** assume we know $Y_1, \ldots, Y_t$
  \[ Y_{t+1} = X_{t+1} + Y_t + Y_{t-15} - Y_{t-16} \]

  Given the past up to time $t$, this is random with distribution $F()$

- **Assume $X \sim iid F()$ with zero mean,**
  the mean of $Y_{t+1}$ given the past up to time $t$ is (point prediction)
  \[ \hat{Y}_t(1) = Y_t + Y_{t-15} - Y_{t-16} \]
Point Predictions

Prediction at lag $\ell = 2$:
assume we know $Y_1, \ldots, Y_t$

$$Y_{t+2} = X_{t+2} + Y_{t+1} + Y_{t-14} - Y_{t-15}.$$ 

Therefore: (point prediction at lag 2)

$$\hat{Y}_t(2) = \hat{Y}_t(1) + Y_{t-14} - Y_{t-15}$$

At lag $\ell$:
use the formula

$$Y_{t+\ell} = X_{t+\ell} + Y_{t+\ell-1} + Y_{t+\ell-16} + Y_{t+\ell-17}$$
in which you replace

$Y_{t+s}$ by $\hat{Y}_t(s)$ for $s > 0$ and $X_{t+\ell}$ by 0 (= the mean of $F()$)

Given the past up to time $t$, the conditional expectation is 0 ($F()$ has zero mean)
PROPOSITION 5.1. Assume that $X = LY$ where $L$ is a differencing or de-seasonalizing filter with impulse response $g_0 = 1, g_1, \ldots, g_q$. Assume that we are able to produce a point prediction $\hat{X}_t(\ell)$ for $X_{t+\ell}$ given that we have observed $X_1$ to $X_t$. For example, if the differenced data can be assumed to be iid with mean $\mu$, then $\hat{X}_t(\ell) = \mu$.

A point prediction for $Y_{t+\ell}$ can be obtained iteratively by:

\[
\hat{Y}_t(\ell) = \hat{X}_t(\ell) - g_1\hat{Y}_t(\ell - 1) - \ldots - g_{\ell-1}\hat{Y}_t(1) - g_{\ell}y_t - \ldots - g_qy_{t-q+\ell} \quad \text{for } 1 \leq \ell \leq q \tag{5.14}
\]

\[
\hat{Y}_t(\ell) = \hat{X}_t(\ell) - g_1\hat{Y}_t(\ell - 1) - \ldots - g_q\hat{Y}_t(\ell - q) \quad \text{for } \ell > q \tag{5.15}
\]
Use of the alternative representation (MA representation of $L^{-1}$)

- Prediction at lag $\ell = 1$: assume we know $Y_1, \ldots, Y_t$

$$Y_{t+1} = X_{t+1} + X_t + \cdots + X_{t-14} + 2(X_{t-15} + X_{t-16} + \cdots + X_{t-30}) + 3(X_{t-31} + X_{t-32} + \cdots + X_{t-46}) + \cdots$$

Given the past up to time $t$, this is random with distribution $F()$

Therefore $\hat{Y}_t(1) = X_t + \cdots + X_{t-14} + 2(X_{t-15} + X_{t-16} + \cdots + X_{t-30}) + 3(X_{t-31} + X_{t-32} + \cdots + X_{t-46}) + \cdots$

- Note it would not be a good idea to use this formula to compute $\hat{Y}_t(1)$ because we accumulate a large number of errors – but it can be used to compute prediction intervals
Computation of Prediction Intervals
(example with $\ell = 3$)

Prediction at lag $\ell = 3$: assume we know $Y_1, \ldots, Y_t$

Since the filter $L$ is causal and invertible, knowing $Y_1, \ldots, Y_t$ is equivalent to knowing $X_1, \ldots, X_t$

\[
Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + X_t + \cdots + X_{t-12} \\
+ 2(X_{t-13} + X_{t-14} + \cdots + X_{t-28}) \\
+ 3(X_{t-39} + X_{t-30} + \cdots + X_{t-44}) + \cdots
\]

Therefore

\[
Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + \hat{Y}_t(3)
\]

(innovation formula)
\[ Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + \hat{Y}_t(3) \]

Given the past up to time \( t \), the distribution of \( Y_{t+3} \) is given by
- a constant \( \hat{Y}_t(3) \)
- plus the sum of 3 independent random variables each with distribution \( F() \) (the assumed distribution of \( X_t \))

Example: assume \( X_t \sim iid N(0, \sigma^2) \)
the distribution of \( X_{t+3} + X_{t+2} + X_{t+1} \) is \( N(0, 3\sigma^2) \)
i.e. the distribution of \( Y_{t+3} \) given the past up to time \( t \) is normal with mean \( \hat{Y}_t(3) \) and variance \( 3\sigma^2 \)
A 95%-prediction interval at lag 3 is...

A. $Y_t(3) \pm 1.96 \sigma$
B. $\hat{Y}_t(3) \pm 1.96 \times \sqrt{3} \sigma$
C. $\hat{Y}_t(3) \pm 1.96 \times 3 \sigma$
D. $\hat{Y}_t(3) \pm 1.96 \times 3 \frac{\sigma}{\sqrt{n}}$
E. None of the above
Solution

The distribution of $Y_{t+3}$ given the past up to time $t$ is normal with mean $\hat{Y}_t(3)$ and variance $3\sigma^2$, therefore with proba 95%, $Y_{t+3}$ is in the interval $\hat{Y}_t(3) \pm 1.96 \times \sqrt{3} \sigma$

Answer B
Prediction assuming differenced data is iid $N(0, \sigma^2)$

Figure 6.7: Differencing filters $\Delta_1$ and $\Delta_{16}$ applied to Example 6.1 (first terms removed). The forecasts are made assuming the differenced data is iid gaussian with 0 mean. $\circ$ = actual value of the future (not used for fitting the model).
PROPOSITION 5.2. Assume that the differenced data is iid gaussian. i.e. \( X_t = (LY)_t \sim \text{iid } N(\mu, \sigma^2) \).

The conditional distribution of \( Y_{t+\ell} \) given that \( Y_1 = y_1, \ldots, Y_t = y_t \) is gaussian with mean \( \hat{Y}_t(\ell) \) obtained from Eq.(5.14) and variance

\[
\text{MSE}_t^2(\ell) = \sigma^2 \left( h_0^2 + \cdots + h_{\ell-1}^2 \right) \tag{5.16}
\]

where \( h_0, h_1, h_2, \ldots \) is the impulse response of \( L^{-1} \). A prediction interval at level 0.95 is thus

\[
\hat{Y}_t(\ell) \pm 1.96 \sqrt{\text{MSE}_t^2(\ell)} \tag{5.17}
\]
Compare the Two

Linear Regression with 3 parameters + variance  
Assuming differenced data is iid
9. Using ARMA Models for the Noise

- This technique is used when the differenced data appears stationary but not iid – the correlation structure can be used to gain some information about futures.
- The differenced data can be modelled as an ARMA process instead of iid.

(c) Differencing at Lags 1 and 16
Deciding whether a stationary $X_t$ is iid

**Sample ACF**

A means to test whether a data series that appears to be stationary is iid or not is the sample autocovariance function: by analogy to the autocovariance of a process, it is defined, for $t \geq 0$ by

$$
\hat{\gamma}_t = \frac{1}{n} \sum_{s=1}^{n-t} (X_{s+t} - \bar{X})(X_s - \bar{X})
$$

(5.20)

where $\bar{X}$ is the sample mean. The sample ACF is defined by $\hat{\rho}_t = \hat{\gamma}_t / \hat{\gamma}_0$. The sample PACF is also defined as an estimator of the true partial autocorrelation function (PACF) defined in Section 5.5.2.

If $X_1, \ldots, X_n$ is iid with finite variance, then the sample ACF and PACF are asymptotically centered normal with variance $1/n$. ACF and PACF plots usually display the bounds $\pm 1.96/\sqrt{n}$. If the sequence is iid with finite variance, then roughly 95% of the points should fall within the bounds. This provides a method to assess whether $X_t$ is iid or not. If yes, then no further modelling is required, and we are back to the case in Section 5.4.2. See Figure 5.10 for an example.

The ACF can be tested formally by means of the Ljung-Box test. It tests $H_0$: “the data is iid” versus $H_1$: “the data is stationary”. The test statistic is $L = n(n + 2) \sum_{s=1}^{t} \frac{\hat{\rho}_s^2}{n-s}$, where $t$ is a parameter of the test (number of coefficients), typically $\sqrt{n}$. The distribution of $L$ under $H_0$ is $\chi_t^2$, which can be used to compute the $p$-value.
Figure 6.10: First panel: Sample ACF of the internet traffic of Figure 6.1. The data does not appear to come from a stationary process so the sample ACF cannot be interpreted as estimation of a true ACF (which does not exist). Second panel: sample ACF of data differenced at lags 1 and 16. The sampled data appears to be stationary and the sample ACF decays fast. The differenced data appears to be suitable for modelling by an ARMA process.
**ARMA Process**

**Definition 6.5.1.** A 0-mean ARMA\((p, q)\) process \(X_t\) is a process that satisfies for \(t = 1, 2, \ldots\) a difference equation such as:

\[
X_t + A_1 X_{t-1} + \cdots + A_p X_{t-p} = \epsilon_t + C_1 \epsilon_{t-1} + \cdots + C_q \epsilon_{t-q} \quad \epsilon_t \text{ iid } \sim N_0, \sigma^2 \quad (6.21)
\]

Unless otherwise specified, we assume \(X_{-p+1} = \cdots = X_0 = 0\).

An ARMA\((p, q)\) process with mean \(\mu\) is a process \(X_t\) such that \(X_t - \mu\) is a 0 mean ARMA process and, unless otherwise specified, \(X_{-p+1} = \cdots = X_0 = \mu\).

The parameters of the process are \(A_1, \ldots, A_p\) (auto-regressive coefficients), \(C_1, \ldots, C_q\) (moving average coefficients) and \(\sigma^2\) (white noise variance). The iid sequence \(\epsilon_t\) is called the noise sequence, or innovation.

An ARMA\((p, 0)\) process is also called an **Auto-regressive** process, AR\((p)\); an ARMA\((0, q)\) process is also called a **Moving Average** process, MA\((q)\).

\[
X = \mu + F \epsilon
\]

\[
F = \frac{1 + C_1 B + \cdots + C_q B^q}{1 + A_1 B + \cdots + A_p B^p}
\]
HYPOTHESIS 6.1. The filter in Eq.(6.23) and its inverse are stable.

In practice, this means that the zeroes of \(1 + A_1z^{-1} + \ldots + A_pz^{-p}\) and of \(1 + C_1z^{-1} + \ldots + C_qz^{-q}\) are within the unit disk.

(a) ARMA(2,2) \(X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}\)

(b) AR(2) \(X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t\)

(c) MA(2) \(X_t = \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}\)

Figure 6.8: Simulated ARMA processes with 0 mean and noise variance \(\sigma^2 = 1\). The first one, for example, is obtained by the matlab commands \(Z=randn(1,n)\) and \(X=filter([1 -0.4 0.95],[1 0.4 -0.45],Z)\).
Which of these matlab scripts produce a sample \( X \) of an ARMA process?

A. \( X=\text{filter}([1;-0.4],[1;0.4],\text{randn}(1,n)) \)
B. \( X=\text{filter}([1;0.4],[1;-0.4],\text{randn}(1,n)) \)
C. A and B
D. None
E. I don’t know
Solution

A and B each produce a sample of $F(\epsilon)$ where $\epsilon$ is an iid sequence of $n$ standard normal random variables and $F$ is an ARMA filter.

We need to verify whether the filter and its inverse are stables. For A, the roots of the numerator polynomial are $z - 0.4 = 0$, thus $z = 0.4$; for the denominator polynomial we have $z + 0.4 = 0$ thus $z = -0.4$ thus both $F$ and $F^{-1}$ are stable.

Idem for B.

Both A and B Are ARMA processes.
Answer C
ARMA Processes are Gaussian (non iid)

**ARMA Process as a Gaussian Process** Since an ARMA process is defined by linear transformation of a gaussian process $\varepsilon_t$ it is a gaussian process. Thus it is entirely defined by its mean $\mathbb{E}(X_t) = \mu$ and its covariance. Its covariance can be computed in a number of ways, the simplest is perhaps obtained by noticing that

$$X_t = \mu + h_0 \varepsilon_t + \ldots + h_{t-1} \varepsilon_1$$

(6.24)

where $h$ is the impulse response of the filter in Eq.(6.23). Note that, with our convention, $h_0 = 1$. It follows that for $t \geq 1$ and $s \geq 0$:

$$\text{cov}(X_t, X_{t+s}) = \sigma^2 \sum_{j=0}^{t-1} h_j h_{j+s}$$

(6.25)

$$\text{cov}(X_t, X_{t+s}) \approx \gamma_s = \sigma^2 \sum_{j=0}^{\infty} h_j h_{j+s}$$

The convergence of the latter series follows from the assumption that the filter is stable. Thus, for large $t$, the covariance does not depend on $t$. More formally, one can show that an ARMA process with Hypothesis 6.1 is asymptotically stationary [5, 33], as required since we want to model stationary data.
\[ \text{var}(X_t) \approx \sigma^2 \sum_{j=0}^{\infty} h_j^2 = \sigma^2 (1 + \sum_{j=1}^{\infty} h_j^2) \geq \sigma^2 \]

The **Auto-Correlation Function** (ACF) is defined as \( \rho_t = \gamma_t / \gamma_0 \). The ACF is quantifies departure from an iid model; indeed, for an iid sequence (i.e. \( h_1 = h_2 = \ldots = 0 \)), \( \rho_t = 0 \) for \( t \geq 1 \). The ACF can be computed from Eq.(6.26) but in practice there are more efficient methods that exploit Eq.(6.23), see [36], and which are implemented in standard packages. One also sometimes uses the **Partial Auto-Correlation Function** (PACF), which is defined in Section A.5.2 as the residual correlation of \( X_{t+s} \) and \( X_t \), given that \( X_{t+1}, \ldots, X_{t+s-1} \) are known.
(a) ARMA(2,2) \( X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2} \)

(b) AR(2) \( X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t \)

(c) MA(2) \( X_t = \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2} \)
ARIMA Process

- $Y_t$ is called an ARIMA process if $X = LY$ is an ARMA process, where $L$ is a combination of differencing and deseasonalizing filters.

- How to fit an ARIMA process?
  - Apply differencing filters until appears stationary
  - Fit the differenced process $X = LY$ using the ARMA fitting procedure (Thm 5.2, Matlab’s armax);
  - Check ACF of residuals; residuals are $\varepsilon_t = X_t - \hat{X}_{t-1}$ (innovation formula)
  - Be careful with overfitting problem – use AIC or BIC; ACF of $X$ may give an idea of order
Fitting an ARMA process is a non-linear optimization problem

- Usually solved by iterative, heuristic algorithms, may converge to a local maximum, may not converge

- Some simple, non MLE, heuristics exist for AR or MA models
  - Ex: fit the AR model that has the same theoretical ACF as the sample ACF

- Common practice is to bootstrap the optimization procedure by starting with a “best guess”
  - AR or MA fit, using heuristic above
Example 6.3: Internet Traffic, continued. The differenced data in Figure 6.10 appear to be stationary and has decaying ACF. We model it as a 0 mean ARMA\((p, q)\) process with \(p, q \leq 20\) and fit the models to the data. The resulting models have very small coefficients \(A_m\) and \(C_m\) except for \(m\) close to 0 or above to 16. Therefore we re-fit the model by forcing the parameters such that

\[
A = (1, A_1, \ldots, A_p, 0, \ldots, 0, A_{16}, \ldots, A_{16+p})
\]

\[
C = (1, C_1, \ldots, C_p, 0, \ldots, 0, C_{16}, \ldots, C_{16+q})
\]

for some \(p\) and \(q\). The model with smallest AIC in this class is for \(p = 1\) and \(q = 3\).
Forecasting with an ARIMA Process $Y_t$

By composition of filters, $Y = L^{-1}X = L^{-1}F\epsilon$ where $F$ is the filter of the ARMA process and $L$ is the differencing filter. Using the impulse response of $L^{-1}F$ and its inverse we obtain formulas similar to those we saw previously. See Prop 5.4 and forecast-exercise.

![Graphs](image.png)

Figure 6.7: Differencing filters $\Delta_1$ and $\Delta_{16}$ applied to Example 6.1 (first terms removed). The forecasts made assuming the differenced data is iid gaussian with 0 mean. o = actual value of the future ($n$ for fitting the model).
Improve Confidence Interval If Residuals are not Gaussian (but appear to be iid)

- Assume residuals are not gaussian but are iid
- How can we get prediction intervals?
- Bootstrap by sampling from residuals
Algorithm 1 Monte-Carlo computation of prediction intervals at level $1 - \alpha$ for time series $Y_t$ using re-sampling from residuals. We are given: a data set $Y_t$, a differencing and de-seasonalizing filter $L$ and an ARMA filter $F$ such that the residual $\epsilon = F^{-1}L Y_t$ appears to be iid; the current time $t$, the prediction lag $\ell$ and the confidence level $\alpha$. $r_0$ is the algorithm's accuracy parameter.

1: $R = \lceil 2 r_0/\alpha \rceil - 1$  \hspace{1cm} \triangleright \text{For example } r_0 = 25, R = 999$
2: compute the differenced data $(x_1, \ldots, x_t) = L(y_1, \ldots, y_t)$
3: compute the residuals $(e_q, \ldots, e_t) = F^{-1}(x_q, \ldots, x_t)$ where $q$ is an initial value chosen to remove initial inaccuracies due to differencing or de-seasonalizing (for example $q =$ length of impulse response of $L$)
4: for $r = 1 : R$ do
5: \hspace{1cm} draw $\ell$ numbers with replacement from the sequence $(e_q, \ldots, e_t)$ and call them $e_{t+1}^r, \ldots, e_{t+\ell}^r$
6: \hspace{1cm} let $e^r = (e_q, \ldots, e_t, e_{t+1}^r, \ldots, e_{t+\ell}^r)$
7: \hspace{1cm} compute $X_{t+1}^r, \ldots, X_{t+\ell}^r$ using $(x_q, \ldots, x_t, X_{t+1}^r, \ldots, X_{t+\ell}^r) = F(e^r)$
8: \hspace{1cm} compute $Y_{t+1}^r, \ldots, Y_{t+\ell}^r$ using Proposition 6.4.1 (with $X_{t+s}^r$ and $Y_{t+s}^r$ in lieu of $\hat{X}_t(s)$ and $\hat{Y}_t(s)$)
9: end for
10: $(Y_{(1)}, \ldots, Y_{(R)}) = \text{sort } (Y_{t+\ell}^1, \ldots, Y_{t+\ell}^R)$
11: Prediction interval is $[Y_{(r_0)}; Y_{(R+1-r_0)}]$
With gaussian assumption

With bootstrap from residuals
10. Other

We have seen a few forecasting recipes
  regression models
  use of differencing filters to make noise stationary
  use of ARMA models to make noise iid
  use of bootstrap

This can be combined or extended. For example: linear regression with ARMA noise
Assume a linear regression model

\[ Y_t = \sum_i \beta_i x_t^i + \epsilon_t \]

where we find that \( \epsilon_t \) does not look iid at all. We can model \( \epsilon_t \) as an ARMA process and obtain

\[ Y_t = \sum_i \beta_i x_t^i + Fw_t \]

where \( F \) is an ARMA filter and \( w_t \) is iid \( N(0, \sigma^2) \)

Apply the inverse filter and obtain a linear regression model

\[ (F^{-1}Y)_t = \sum_i \beta_i (F^{-1}x_t^i) + w_t, \quad \text{with} \quad w_t \sim \text{iid } N(0, \sigma^2) \]

If we know \( F \) we can estimate \( \beta \); if we know \( \beta \) we can estimate \( F \) \( \Rightarrow \) iterate and hope it converges

Prediction formulae can be obtained using the calculus of filters exactly as we did above.
Sparse ARMA Models

- Problem: avoid many parameters when the degree of the A and C polynomials are high
- Based on heuristics
  - Multiplicative ARIMA, constrained ARIMA
  - Holt Winters

See section 5.6
Sparse models give less accurate predictions but have much fewer parameters and are simple to fit.