Property of acf-hat for a slowly trending series


Exhibit 6.18  Sample ACF for the Oil Price Time Series

Discuss
Empirical model building

Using EDA on data

and on results of arma(p,d,q) fit

Fitting arima is an important method for final/paper analyses.

AIC and BIC are tools for estimating (p,q) of arma(p,q)

\[
\text{AIC} = -2\log(\text{maximum likelihood}) + 2k \\
\text{BIC} = -2\log(\text{maximum likelihood}) + k\log(n)
\]
Example. Canadian hare.

Exhibit 1.5 Abundance of Canadian Hare

Notice low values in 1928, 1929, 1930
Is there a better model (square root found by Box-Cox) than AR(3)?

Evaluate AIC for p=0.5, q=0.6

Look for (p,q) corresponding to smallest value

Table of AIC values

<table>
<thead>
<tr>
<th>p = 0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>q = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>143.86</td>
<td>121.74</td>
<td>102.91</td>
<td>101.08</td>
<td>102.94</td>
<td>103.52</td>
<td></td>
</tr>
<tr>
<td>119.50</td>
<td>114.86</td>
<td>102.33</td>
<td>106.29</td>
<td>101.86</td>
<td>101.30</td>
<td>1</td>
</tr>
<tr>
<td>111.88</td>
<td>110.11</td>
<td>98.55</td>
<td>100.55</td>
<td>101.61</td>
<td>104.41</td>
<td>2</td>
</tr>
<tr>
<td>102.94</td>
<td>103.63</td>
<td>100.54</td>
<td>99.82</td>
<td>101.03</td>
<td>102.94</td>
<td>3</td>
</tr>
<tr>
<td>103.99</td>
<td>104.72</td>
<td>100.70</td>
<td>100.12</td>
<td>102.09</td>
<td>104.04</td>
<td>4</td>
</tr>
<tr>
<td>101.61</td>
<td>105.04</td>
<td>102.02</td>
<td>102.09</td>
<td>104.08</td>
<td>104.78</td>
<td>5</td>
</tr>
<tr>
<td>100.76</td>
<td>100.51</td>
<td>101.54</td>
<td>103.43</td>
<td>104.14</td>
<td>104.89</td>
<td>6</td>
</tr>
</tbody>
</table>

Call:

```r
arima(x = Hare[, 2], order = c(2, 0, 2), method = "ML")
```

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ar2</th>
<th>ma1</th>
<th>ma2</th>
<th>intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1968</td>
<td>-0.7591</td>
<td>0.0216</td>
<td>1.0000</td>
<td>5.7801</td>
</tr>
</tbody>
</table>

s.e. 0.1383 0.1315 0.1211 0.3152 0.5604

sigma^2 estimated as 0.7692: log likelihood = -44.28, aic = 98.55

Fitted model: Y_t = 5.78 + 1.20 Y_{t-1} - 0.76 Y_{t-2} + e_t - 0.02 e_{t-1} - 1.00 e_{t-2}
Need to examine model validity: gaussianity, outliers, …

\{e_t\} \text{ errors/innovations} \quad \text{new information}

\texttt{arima()} \text{ in TSA and elsewhere assumes errors/innovations Gaussian}
qqplot of innovation error estimates

qqplot(arima22.out$res)

innovation outlier!

detectIO(arima22.out)

[,1]
ind 24.000000
lambda1 -3.515447
Remember the low values in the figure

detectAO(arima22.out)

[1] "No AO detected"

eacf() the extended autocorrelation function

Commands residuals(), fit()
The EACF method uses the fact that if the AR part of a mixed ARMA model is known, "filtering out" the autoregression from the observed time series results in a pure MA process that enjoys the cutoff property in its ACF. The AR coefficients may be estimated

\[ 1.96 \sqrt{\frac{n-j-k}{n}} \text{ since the sample autocorrelation is asymptotically } N(0,1/(n-k-j)) \text{ if the } W \text{'s are approximately an MA}(j) \text{ process) and 0 otherwise. In such a table, an } MA(p,q) \text{ process will have a theoretical pattern of a triangle of zeroes, with the upper left-hand vertex corresponding to the ARMA orders. Exhibit 6.4 displays the schematic}

Exhibit 6.4  Theoretical Extended ACF (EACF) for an ARMA(1,1) Model

<table>
<thead>
<tr>
<th>AR/MA</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> eacf.out=eacf(hare,ar.max=5,ma.max=6)

AR/MA
  0 1 2 3 4 5 6
0 x o o x x x o
1 x o o o o x x
2 o o o o o o o o
3 o o o o o o o o
4 o o o o o o o o
5 x o o o o o o o

Add lines starting at (2,1)

p = 2, q = 1
Stationarity

General Electric stock stationary?

Dickey-Fuller (DF)

tests whether a unit root is present in an autoregressive model.

The augmented Dickey–Fuller (ADF) statistic, used in the test, is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence. It is an augmented version of the DF test for a larger and more complicated set of time series models.
Augmented Dickey-Fuller Test

Description:

Computes the Augmented Dickey-Fuller test for the null that \( x \) has a unit root.

Usage:

```
adf.test(x, alternative = c("stationary", "explosive"),
    k = trunc((length(x)-1)^{(1/3)}))
```

Arguments:

- `x`: a numeric vector or time series.
- `alternative`: indicates the alternative hypothesis and must be one of "stationary" (default) or "explosive". You can specify
Consider the rwalk series and its diff series

top plot rwalk, bottom diff(rwalk)
library(tseries)

> data(rwalk)

> adf.test(rwalk,"stationary")

Augmented Dickey-Fuller Test
data:  rwalk  
Dickey-Fuller = -3.9699, Lag order = 3, p-value = 0.01701
alternative hypothesis H_A: stationary

H_0: unit root exists

H_0 rejected at 5% level, not at 1%

Difference case

> adf.test(diff(rwalk),"stationary")

Augmented Dickey-Fuller Test
data:  diff(rwalk)
Dickey-Fuller = -4.0538, Lag order = 3, p-value = 0.01343
alternative hypothesis hypothesis $H_A$: stationary

$H_0$: unit root exists

$H_0$ rejected at 5% level, not at 1%

Returning to the GE series

```
adf.test(ge[,2],"stationary")
Augmented Dickey-Fuller Test
```

data:  ge[, 2]

Dickey-Fuller = -8.7603, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

[Warning message:
In adf.test(ge[, 2], "stationary") : p-value smaller than printed p-value]

Alternative hypothesis, $H_A$: stationary

$H_0$: unit root exists

$H_0$ is rejected at 1% level
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acf</td>
<td>Computes and plots the sample autocorrelation function starting with lag 1.</td>
</tr>
<tr>
<td>arima</td>
<td>This command has been amended to compute the AIC according to our definition.</td>
</tr>
<tr>
<td>arima.boot</td>
<td>Bootstraps time series according to a fitted ARMA($p,d,q$) model.</td>
</tr>
<tr>
<td>arimax</td>
<td>Extends the arima function, allowing the incorporation of transfer functions and innovative and additive outliers.</td>
</tr>
<tr>
<td>ARMAspec</td>
<td>Computes and plots the theoretical spectrum of an ARMA model.</td>
</tr>
<tr>
<td>armasubsets</td>
<td>Finds “best subset” ARMA models.</td>
</tr>
<tr>
<td>BoxCox.ar</td>
<td>Finds a power transformation so that the transformed time series is approximately an AR process with normal error terms.</td>
</tr>
<tr>
<td>detectAO</td>
<td>Detects additive outliers in time series.</td>
</tr>
<tr>
<td>detectIO</td>
<td>Detects innovative outliers in time series.</td>
</tr>
<tr>
<td>eacf</td>
<td>Computes and displays the extended autocorrelation function of a time series.</td>
</tr>
<tr>
<td>garch.sim</td>
<td>Simulates a GARCH process.</td>
</tr>
<tr>
<td>gBox</td>
<td>Performs a goodness-of-fit test for fitted GARCH models.</td>
</tr>
<tr>
<td>harmonic</td>
<td>Creates a matrix of the first $m$ pairs of harmonic functions for fitting a harmonic trend (cosine-sine trend, Fourier regression) model with a time series response.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Keenan.test</td>
<td>Carries out Keenan's test for nonlinearity against the null hypothesis that the time series follows some AR process.</td>
</tr>
<tr>
<td>kurtosis</td>
<td>Calculates the (excess) coefficient of kurtosis.</td>
</tr>
<tr>
<td>lagplot</td>
<td>Computes and plots nonparametric regression functions of a time series against its various lags.</td>
</tr>
<tr>
<td>periodogram</td>
<td>Computes the periodogram of a time series.</td>
</tr>
<tr>
<td>LB.test</td>
<td>Computes the Ljung-Box or Box-Pierce tests checking whether or not the residuals from an ARIMA model appear to be white noise.</td>
</tr>
<tr>
<td>McLeod.Li.test</td>
<td>Perform the McLeod-Li test for conditional heteroscedasticity (ARCH).</td>
</tr>
<tr>
<td>plot.Arima</td>
<td>Plots a time series and its predictions (forecasts) with 95% prediction bounds based on a fitted ARIMA model.</td>
</tr>
<tr>
<td>predict.TAR</td>
<td>Calculates predictions based on a fitted TAR model. The errors are assumed to be normally distributed and the predictive distributions are approximated by simulation.</td>
</tr>
<tr>
<td>prewhiten</td>
<td>Bivariate time series are prewhitened according to an AR model fitted to the x-component of the bivariate series. Alternatively, if an ARIMA model is provided, it is used to prewhiten both series. The CCF of the prewhitened bivariate series is then computed and plotted.</td>
</tr>
<tr>
<td>qar.sim</td>
<td>Simulates a first-order quadratic AR model with normally distributed white noise error terms.</td>
</tr>
<tr>
<td>rstandard.Arima</td>
<td>Computes internally standardized residuals from a fitted ARIMA model.</td>
</tr>
<tr>
<td>runs</td>
<td>Tests the independence of a sequence of values by checking whether there are too many or too few runs above (or below) the median.</td>
</tr>
<tr>
<td>season</td>
<td>Extracts season information from a time series and creates a vector of the season information. For example, for monthly data, the function outputs a vector containing the months of the data.</td>
</tr>
<tr>
<td>skewness</td>
<td>Calculates the skewness coefficient of a dataset.</td>
</tr>
<tr>
<td>spec</td>
<td>Allows the user to invoke either the spec.pgram function or the spec.ar function in the stats package. The seasonal attribute of the data, if it exists, is suppressed for our preferred way of presenting the output. Alternating defaults to demean=T, detrend=F, taper=0, and permits plotting of confidence interval bands.</td>
</tr>
<tr>
<td>summary.armasubsets</td>
<td>Summary method for class armasubsets, that is useful for ARMA subset selection.</td>
</tr>
<tr>
<td>tar</td>
<td>Estimates a two-regime TAR model.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>tar.sim</code></td>
<td>Simulates a two-regime TAR model.</td>
</tr>
<tr>
<td><code>tar.skeleton</code></td>
<td>Obtains the skeleton of a TAR model by suppressing the noise term in the TAR model.</td>
</tr>
<tr>
<td><code>tlrt</code></td>
<td>Carries out the likelihood ratio test for threshold nonlinearity, with the null hypothesis being a normal AR process and the alternative hypothesis being a TAR model with homogeneous, normally distributed errors.</td>
</tr>
<tr>
<td><code>Tsay.test</code></td>
<td>Carries out Tsay’s test for quadratic nonlinearity in a time series.</td>
</tr>
<tr>
<td><code>tsdiag.Arima</code></td>
<td>Modifies the <code>tsdiag</code> function of the <code>stats</code> package suppressing initial residuals and displaying Bonferroni bounds. It also checks the condition for the validity of the chi-square asymptotics for the portmanteau tests.</td>
</tr>
<tr>
<td><code>tsdiag.TAR</code></td>
<td>Displays the time series plot and the sample ACF of the standardized residuals. Also, portmanteau tests for detecting autocorrelations in the standardized residuals are computed and displayed.</td>
</tr>
<tr>
<td><code>zlag</code></td>
<td>Computes the lag of a vector, with missing elements replaced by NA.</td>
</tr>
</tbody>
</table>
library(TSA)
par(mfrow=c(2,1))
data(hare)
Hare<-cbind(1905:1935,as.double(hare))
ny<-length(Hare[,1])
plot(Hare[,1],Hare[,2],type="l",ylab="relative abundance",xlab="year",main="Snow hare")
acf(Hare[,2],lag.max=10,xlab="year",type="cor",main="")
title("acf(cor) Canadian hare")
graphics.off()
library(TSA)
data(hare)

Hare <- cbind(1905:1935, as.double(sqrt(hare)))

ny <- length(Hare[,1])
table <- NULL

Q <- 6

for(q in 0:Q){
  P <- 5
  row <- NULL

  for(p in 0:P){
    arima.out <- arima(Hare[,2], order = c(p,0,q), method = "ML")
    row <- c(row, arima.out$aic)
  }

  table <- rbind(table, row)
}

analge
par(mfrow=c(2,1))

ge<-matrix(scan("m-ge2608.txt"),byrow=T,ncol=2)
ge[,1]<-ge[,1]/10000
gef<-floor(ge[,1])
geg<-ge[gef>1939,]
ny<-length(geg[,1])/12
frac<-rep(1:12,ny)
frac1<-(rep(1:12,ny)-.5)/12

type="l",ylab="returb",xlab="year",main="GE stock")
acf(geg[,2],lag.max=60,xlab="month",type="cor",main="")
title("acf(cor) GE stock returns")

graphics.off()
library(TSA)

ge<-matrix(scan("m-ge2608.txt"),byrow=T,,ncol=2)

ge[,1]<-ge[,1]/10000

gef<-floor(ge[,1])

geg<-ge[gef>1939,]

ny<-length(geg[,1])/12

frac<-rep(1:12,ny)

frac1<-(rep(1:12,ny)-.5)/12

geg[,1]<-geg[,1]+frac1

tablege<-NULL

Q<-6

for(q in 0:Q){

P<-5

row<-NULL

for(p in 0:P){

arima.out<-arima(geg[,2],order=c(p,0,q),method="ML")

row<-c(row,arima.out$aic)

}

tablege<-rbind(tablege,row)

}