HG

Third lecture - 29. Jan. 2014

1. Joint stationarity and long run effects in a simple ADL(1,1)

Suppose $\{X_t\},\{Y_t\}$ are two stationary time series.

Does it follow that the sum, $\{X_t + Y_t\}$, also is stationary?

The answer is NO in general. There may be non-stationarity present in the covariance structure between the two series.

On the other hand, if we require that $\{X_t\},\{Y_t\}$ are *jointly stationary*, then $\{X_t + Y_t\}$ is stationary.

DEF.1 (C.f. Hamilton section 10.2) Consider the vector time series $\{V_t\} = \begin{Bmatrix} X_t \\ Y_t \end{Bmatrix}$. Then we define *joint (covariance) stationarity* for $\{X_t\},\{Y_t\}$, to mean that the vector time series $\{V_t\}$ is *covariance stationary* – meaning that

i. $EV_t = \begin{Bmatrix} EX_t \\ EY_t \end{Bmatrix} = \begin{Bmatrix} \mu_x \\ \mu_y \end{Bmatrix}$ is constant, and

ii.

$$\Sigma(h) = \text{cov}(V_t, V_{t+h}) = \begin{pmatrix} \text{cov}(X_t, X_{t+h}) & \text{cov}(X_t, Y_{t+h}) \\ \text{cov}(Y_t, X_{t+h}) & \text{cov}(Y_t, Y_{t+h}) \end{pmatrix} =$$

$$= \begin{pmatrix} \gamma_x(h) & \gamma_{xy}(h) \\ \gamma_{yx}(h) & \gamma_y(h) \end{pmatrix}, \quad h = 0, \pm 1, \pm 2$$

does not depend on *t*.

Note that, in contrast to univariate time series where the autocovariance satisfies $\gamma(-h) = \gamma(h)$, we do not have the same property for vector time series, i.e., in general, $\Sigma(-h) \neq \Sigma(h)$ - due to the fact that, in general, we may have $\gamma_{xy}(h) \neq \gamma_{yx}(h)$.

If the vector time series $\{V_t\}$ is covariance stationary, it follows that any linear combination $\{aX_t + bY_t\}$ is stationary as well.

Now suppose $\{X_t\},\{Y_t\}$ are jointly stationary and satisfy the following dynamic specification.

(1) $Y_t = \varphi_0 + \varphi_1 Y_{t-1} + \beta_0 X_t + \beta_1 X_{t-1} + \varepsilon_t$, where $\varepsilon_t \sim WN(0,\sigma^2)$

We assume that

(i) ε_t is an *innovation* - meaning that ε_t is uncorrelated with the prehistory of Y_t, i.e., $Y_{t-1}, Y_{t-2}, \ldots$. This can be achieved by requiring that the solution of $\{Y_t\}$ from (1) should be causal, i.e., requiring that $|\varphi_1| < 1$, and

(ii) that $\{X_t\}$ is a causal stationary (i.e., that $|\beta_1/\beta_0| < 1$), and

(iii) that $\{X_t\}$ is exogenous in the strong sense that ε_t is independent of the whole series $\{X_t\}$.

This is a special case of an *autoregressive distributed lag model* of order 1 and 1 (an ADL(1,1) model).

Note.

- Such a model can be estimated by simple OLS which provides consistent estimates for all 5 parameters. In this context $Y_{t-1}, Y_{t-2}, \ldots$ are called *predetermined*, which plays the role of a weaker exogeneity assumption ensuring that OLS works - treating Y_{t-1} as an explanatory variable.
- If the error ε_t is replaced by a MA error, for example by $u_t = \varepsilon_t + \theta\varepsilon_{t-1}$, then Y_{t-1} can no longer be treated as an explanatory variable in (1) since it will be correlated with the error, u_t. Also (1) is no longer an ADL model, but a more general ARMAX model that needs other estimation methods than OLS.

With these assumptions we can easily find the long-run effects involved here. There are at least two ways of thinking about this:

Method A:
Find the long-run effects from the (causal) solution of (1), which exists since $|\varphi_1| < 1$. Write first (1) in terms of lag polynomials

(2) $\varphi(L)Y_t = \varphi_0 + \beta(L)X_t + \varepsilon_t$, where $\varphi(L) = 1 - \varphi_1 L$ and $\beta(L) = \beta_0 + \beta_1 L$

Now, since $|\varphi_1| < 1$, the infinite order linear filter $1/\varphi(L)$ is well defined, and we find the solution of (2) simply as

$$(3) \qquad Y_t = \frac{1}{\varphi(L)}\varphi_0 + \frac{\beta(L)}{\varphi(L)} X_t + \frac{1}{\varphi(L)}\varepsilon_t = \frac{\varphi_0}{1-\varphi_1} + \frac{\beta(L)}{\varphi(L)} X_t + \frac{1}{\varphi(L)}\varepsilon_t =$$

From this we see that we have 2 long-run effects here, (1) the effect of a unit change in ε_t leaving all other ε_s unchanged. We found in LN2 that this is $1/\varphi(1) = \frac{1}{1-\varphi_1}$. Similarly (2), since $\frac{\beta(L)}{\varphi(L)} = \delta(L) = \delta_0 + \delta_1 L + \delta_2 L^2 + \cdots$ the long-run effect on $\{Y_t\}$ of a unit change in X_t leaving all other X_s unchanged,

$$\sum_{j=0}^{\infty} \delta_j = \delta(1) = \frac{\beta(1)}{\varphi(1)} = \frac{\beta_0 + \beta_1}{1-\varphi_1}$$

Method B: (quite common in econometrics). Here we imagine that the dynamic specification in (1) implies *a long-run steady state* obtained when all errors (including those for X_t) are set = 0. If the errors are set to zero, both X_t and Y_t will, due to stationarity, take the expected values, μ_x and μ_y respectively, and the steady state relationship becomes,

$$\mu_y = \varphi_0 + \varphi_1\mu_y + \beta_0\mu_x + \beta_1\mu_x \quad \Leftrightarrow$$
$$(1-\varphi_1)\mu_y = \varphi_0 + (\beta_0 + \beta_1)\mu_x \quad \Leftrightarrow$$

$$(4) \qquad \mu_y = \frac{\varphi_0}{1-\varphi_1} + \frac{\beta_0 + \beta_1}{1-\varphi_1}\mu_x$$

so a unit shift in the long-run steady state of X_t will be accompanied by a shift of $\frac{\beta_0 + \beta_1}{1-\varphi_1}$ in the steady state for Y_t.

This section is a pre-taste of the study of relationships between time series presented by Ragnar later.

We return to the univariate case.

2. Autocorrelation function for a causal ARMA(p,q)

ARMA(*p*,*q*):

(5) $\varphi(L)Y_t = \varphi_0 + \theta(L)\varepsilon_t$, where $\varphi(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p, \quad \theta(L) = (1 + \theta_1 L + \cdots + \theta_q L^q)$

and where $\varepsilon_t \sim WN(0, \sigma^2)$.

We assume that all roots of $\varphi(z)$ are outside the unit circle, in which case (5) has a causal stationary solution, with expected value, $\mu = E(Y_t) = \varphi_0 / \varphi(1)$.

Introducing the centered series, $y_t = Y_t - \mu$, exercise 3 of seminar1 shows that (5) is equivalent with

(6) $\varphi(L)y_t = \theta(L)\varepsilon_t$ without the constant φ_0, and where $E(y_t) = 0$.

The autocovariance function, is now $\gamma(-h) = \gamma(h) = E(y_t y_{t+h}), \quad h = 0,1,2,3,\ldots$
and the autocorrelation function (*acf*), $\rho(-h) = \rho(h) = \gamma(h)/\gamma(0), \quad h = 0,1,2,3,\ldots$
where $\gamma(0) = \text{var}(Y_t) = \text{var}(y_t)$ and $\rho(0) = 1$.

Case 1, *p* = 0 (MA(*q*).
We find, putting $\theta_0 = 1$,

$$\gamma(h) = E(y_t y_{t+h}) = \sigma^2 \left(\sum_{j=0}^{q} \theta_j \varepsilon_{t-j}\right)\left(\sum_{k=0}^{q} \theta_k \varepsilon_{t-k}\right) = \begin{cases} \sigma^2 \sum_{j=0}^{q} \theta_j \theta_{j+h}, & 0 \le h \le q \\ 0 & h > q \end{cases}$$

giving the acf

$$\rho(h) = \begin{cases} \dfrac{\sum_{j=0}^{q} \theta_j \theta_{j+h}}{1 + \theta_1^2 + \cdots + \theta_q^2} & 0 \le h \le q \\ 0 & h > q \end{cases}$$

Case 2, *p* > 0 (ARMA(*p*,*q*)).

The main point is that both $\{\gamma(h)\}$ and $\{\rho(h)\}$ satisfy the same homogeneous difference equation as the dynamic multipliers $\{\psi_j\}$ and $\{\delta_j\}$ we found in LN2, i.e.,

(7) $\varphi(L)\gamma(h)=0$ and $\varphi(L)\rho(h)=0$ for $h\geq \max(p,q+1)$

with slightly different initial values for $h<\max(p,q+1)$.

For completeness sake:
For $\{\gamma(h)\}$, the initial conditions can, in general, be expressed by the θ_j 's and the dynamic multipliers from $\delta(L)=\theta(L)/\varphi(L)=\delta_0+\delta_1 L+\delta_2 L^2+\cdots$

(8)
$$\gamma(h)-\varphi_1\gamma(h-1)-\cdots-\varphi_p\gamma(h-p)=\theta_h\delta_0+\theta_{h+1}\delta_1+\cdots+\theta_q\delta_{q-h}$$
$$\text{for } 0\leq h<\max(p,q+1)$$

(Note that in the AR(p) case $\delta(L)=1/\varphi(L)=\psi(L)=\psi_0+\psi_1 L+\psi_2 L^2+\cdots$)

In the AR(*p*) =ARMA(*p*,0) case (8) reduces to something slightly simpler

(9)
$$\gamma(h)-\varphi_1\gamma(h-1)-\cdots-\varphi_p\gamma(h-p)=\begin{cases}\sigma^2 & \text{for } h=0\\ 0 & \text{for } h\neq 0\end{cases}$$

For small values of *p* and *q* it is not necessary to go through $\{\delta_j\}$ or $\{\psi_j\}$ but easier to calculate the initial values directly (see, e.g., Hamilton at the end of section 3.4 for the AR(2) case.)

Having found $\{\gamma(h)\}$, it is easy to find the acf $\{\rho(h)\}$ using $\rho(-h)=\rho(h)=\gamma(h)/\gamma(0)$.

Since the general solution for $\{\gamma(h)\}$ is equal to the general solution for $\{\psi_h\}$, we see that $\rho(h) = \gamma(h)/\gamma(0)$ approaches 0 just as fast as ψ_h when $h \to \infty$, i.e., exponentially fast.

3. Estimation of an causal ARMA(p,q) model (a rough survey)

Suppose $Y_t \sim \text{ARMA}(p,q)$ such that $\varphi(L)Y_t = \varphi_0 + \theta(L)\varepsilon_t$ that we rewrite in the form (see Ex.3 in seminar 1)

(10) $\varphi(L)(Y_t - \mu) = \theta(L)\varepsilon_t$,

where $\mu = E(Y_t) = \varphi_0/\varphi(1)$, and $\varepsilon_t \sim WN(0,\sigma_\varepsilon^2)$.

We have observations of $Y_1,\ldots,Y_T$ and want to estimate the parameters in (10). There are several more or less asymptocically equivalent estimation procedures available. It turns out that maximum likelihood methods based on Gaussian joint distributions usually work well even if the Y_t 's are not normally distributed. A few methods can be mentioned

- Full maximum likelihood (mle) based on Gaussian distribution,
- Conditional mle based on conditional (Gaussian) joint conditional distribution of $Y_2,\ldots,Y_T$ given fixed start value Y_1.
- Unconditional least squares estimation.
- Conditional least squares given fixed Y_1.
- Yule-Walker estimation in the AR(p) case which are the same as the moment estimators (mme).

If the parameter vector (except σ_ε^2) is $\beta' = (\mu,\varphi_1,\ldots,\varphi_p,\theta_1,\ldots,\theta_q)$, then all these estimators lead to the same asymptotic normal distribution (even if the observations are not Gaussian)

$\sqrt{T}(\hat{\beta} - \beta) \underset{T\to\infty}{\overset{F}{\to}} N(0,\sigma_\varepsilon^2 C)$ where $\sigma_\varepsilon^2 C$ is an asymptotic covariance matrix derived from a Gaussian Fisher information matrix, and which can be consistently estimated. Details (somewhat complicated) can partly be found in Hamilton and the more advanced time series literature (and will not be discussed here).

The default estimation procedure used by Stata is the full mle, but one may order some of the other methods as well.

The full mle method are based on the factorization of the joint Gaussian distribution

$$f(y_1,\dots,y_T) = f_1(y_1)f(y_2 \mid y_1)f(y_3 \mid y_1,y_2)\cdots f(y_T \mid y_1,\dots,y_{T-1})$$

The simple AR(1) case may illustrate the different methods:
Then (in the Gaussian case) , ε_t is independent of the prehistory, $Y_{t-1},Y_{t-2},\dots$ and $Y_t-\mu=\varphi_1(Y_{t-1}-\mu)+\varepsilon_t$ implies that $f(y_t \mid y_1,\dots,y_{t-1}) = f(y_t \mid y_{t-1})$, which is simply the $N(\varphi_1(y_{t-1}-\mu),\sigma_\varepsilon^2)$ density (pdf).
The start, Y_1, has a different normal pdf , $N(\mu,\sigma_y^2)$ with parameter determined by stationarity, $\mu = E(Y_t)$, $\quad \sigma_y^2 = \text{var}(Y_t) = \dfrac{\sigma_\varepsilon^2}{1-\varphi_1^2}$ (the last follows since $Y_t-\mu=\varphi_1(Y_{t-1}-\mu)+\varepsilon_t \Rightarrow \sigma_y^2 = \varphi_1^2\sigma_y^2+\sigma_\varepsilon^2$)
The full log likelihood then becomes

$$\text{(11)} \quad l_{full} = \text{constant} - \frac{T}{2}\ln(\sigma_\varepsilon^2) - \frac{1}{2}\ln(1-\varphi_1^2) - \frac{1}{2\sigma_\varepsilon^2}S(\mu,\varphi_1)$$

where

$$\text{(12)} \quad S(\mu,\varphi_1) = (1-\varphi_1^2)(y_0-\mu)^2 + \sum_{t=2}^{T}\left[(y_t-\mu)-\varphi_1(y_{t-1}-\mu)\right]^2$$

is called **the unconditional sum of squares**.

Full mle (Stata's default) means maximizing (11), which requires iterations, and then estimating $\hat{\sigma}_\varepsilon^2 = \dfrac{S(\hat{\mu},\hat{\varphi}_1)}{T}$.

Unconditional least squares means minimizing (12) first, which also requires iterations, and then estimating $\hat{\sigma}_\varepsilon^2 = \dfrac{S(\hat{\mu},\hat{\varphi}_1)}{T-2}$.

Conditional mle uses the simpler conditional log likelihood

(13) $$l(\mu,\varphi_1,\sigma_\varepsilon^2 \mid y_0) = cons. - \frac{T-1}{2}\ln(\sigma_\varepsilon^2) - \frac{S_c(\mu,\varphi_1)}{2\sigma_\varepsilon^2}$$

where

(14) $$S_c(\mu,\varphi_1) = \sum_{t=2}^{T}\left[(y_t-\mu)-\varphi_1(y_{t-1}-\mu)\right]^2$$

is called **the conditional sum of squares** (being equal to a usual sum of squares).

The conditional mle maximizes (13) and the conditional least squares minimizes (14) giving the estimators

$$\hat{\mu} = \frac{\bar{y}_1 - \hat{\varphi}_1\bar{y}_2}{1-\hat{\varphi}_1}, \text{ where } \bar{y}_1 = \frac{1}{T-1}\sum_{t=1}^{T-1} Y_t \text{ and } \bar{y}_2 = \frac{1}{T-1}\sum_{t=2}^{T} Y_t$$

(showing $\hat{\mu} \approx \bar{Y} = \sum_{t=1}^{T} Y_t \Big/ T$, i.e., the mme = the Yule-Walker estimator)

and $\hat{\varphi}_1 = \dfrac{\sum_{t=2}^{T}(Y_t-\bar{Y}_2)(Y_{t-1}-\bar{Y}_1)}{\sum_{t=2}^{T}(Y_{t-1}-\bar{Y}_1)^2}$, which is quite close to the usual OLS estimator (where we would use the same mean $\bar{Y}$ instead of $\bar{Y}_1,\bar{Y}_2$) .

Note. In principle, using OLS on an AR(p) would give consistent and asymptotic normally distributed estimators for the same reason as given for the ADL model in section 1. However, it was discovered that the OLS estimators as well as the conditional least squares estimators has a tendency of producing a bias, the so called Hurwitz-bias, of size $-\frac{2\varphi_1}{T}$ in moderate samples. This is nicely described and illustrated in Ragnar's first lecture notes 2011. This is probably the main reason why modern software programs usually prefer the full mle estimation to compensate for this bias. The necessity of iterations is hardly an issue any more…)

The Yule-Walker estimator is the moment method based on (9) reproduced here

$$\gamma(h) = \begin{cases} \varphi_1\gamma(h-1) + \cdots + \varphi_p\gamma(h-p) + \sigma_\varepsilon^2 & \text{for } h = 0 \\ \varphi_1\gamma(h-1) - \cdots - \varphi_p\gamma(h-p) & \text{for } h \neq 0 \end{cases}$$

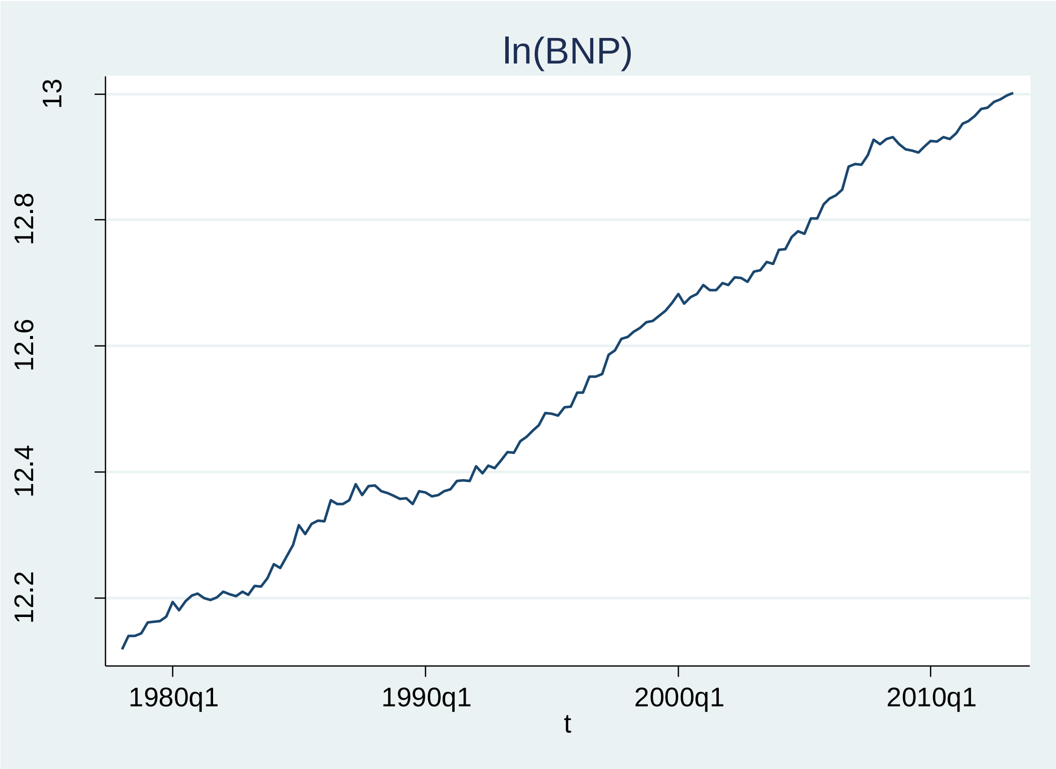
which in matrix notation becomes

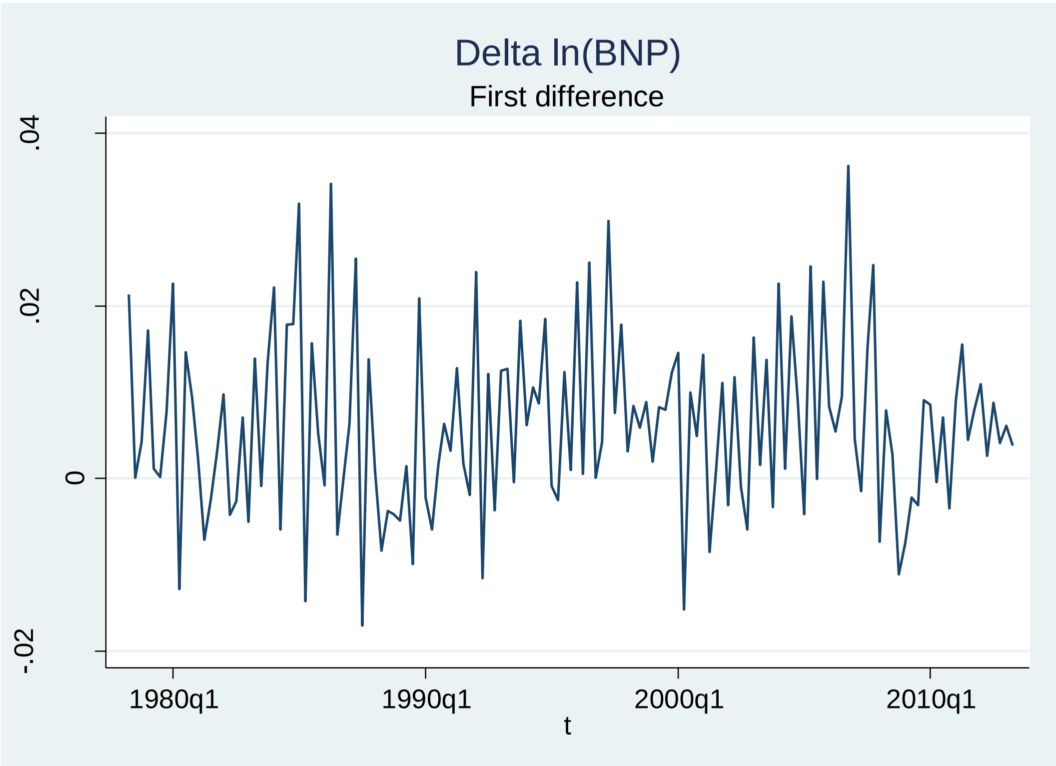
$$\Gamma_p\varphi = \gamma_p, \quad \text{and} \quad \sigma_\varepsilon^2 = \gamma(0) - \varphi'\gamma_p, \text{ where}$$

$\Gamma_p = \{\gamma(k-j)\}_{j,k=1}^{p}$ is a $p \times p$ matrix, $\varphi = (\varphi_1, \ldots, \varphi_p)'$ a $p \times 1$ vector, and $\gamma_p = (\gamma(0), \ldots, \gamma(p))'$ is a $p \times 1$ vector. The method of moments consists in replacing all $\gamma(h)$ by sample estimates $\hat{\gamma}(h)$ and solve

$$\hat{\varphi} = \hat{\Gamma}_p^{-1}\hat{\gamma}_p, \quad \hat{\sigma}_\varepsilon^2 = \hat{\gamma}(0) - \hat{\gamma}_p'\hat{\Gamma}_p^{-1}\hat{\gamma}_p$$

4. A practical example – the BNP data from LN1.





The Box-Jenkins' identification process (described below) leads to the decision that the "best" model appears to be a (causal) ARMA(1,2) process for $\Delta Y_t = \Delta \ln BNP_t$. This we express by saying that $Y_t = \ln BNP_t$ is an ARIMA(1,1,2).

(**DEF:** $X_t \sim \text{ARIMA}(p,d,q)$ means that $\Delta^d X_t \sim \text{ARMA}(p,q)$ (causal))

The estimation of this model is done by Stata, using the `arima` command:

```
. arima y,arima(1,1,2)            (I have defined y = ln(bnp) in my dataset)

(setting optimization to BHHH)
Iteration 0:   log likelihood =  448.92264
Iteration 1:   log likelihood =  453.54759
Iteration 2:   log likelihood =  454.73258
Iteration 3:   log likelihood =  455.93749
Iteration 4:   log likelihood =  456.03335
(switching optimization to BFGS)
Iteration 5:   log likelihood =   456.0458
Iteration 6:   log likelihood =  456.05034
Iteration 7:   log likelihood =  456.05047
Iteration 8:   log likelihood =  456.05049
```

```
ARIMA regression

Sample:  1978q2 - 2013q2                        Number of obs      =       141
                                                Wald chi2(3)       =     96.22
Log likelihood =  456.0505                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
         D.y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   .0062495   .0012077     5.17   0.000     .0038826    .0086165
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .7294206   .1237281     5.90   0.000     .4869179    .9719232
             |
          ma |
         L1. |  -1.104681   .1158547    -9.54   0.000    -1.331752   -.8776096
         L2. |   .4904761   .0843566     5.81   0.000     .3251403     .655812
-------------+----------------------------------------------------------------
      /sigma |    .009512   .0006189    15.37   0.000      .008299     .010725
------------------------------------------------------------------------------
```

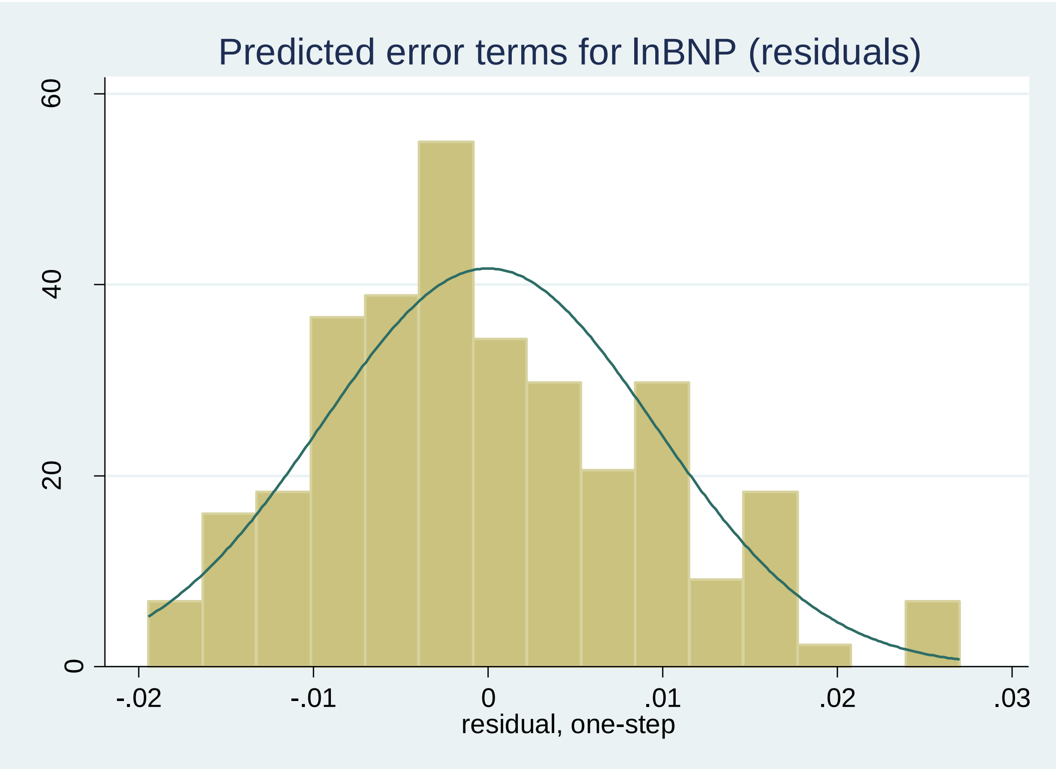
```
Note: The test of the variance against zero is one sided, and the two-sided
confidence interval is
    truncated at zero.
```

Having estimated an ARIMA model one should always check if there is any evidence in the data against the model (specification testing). The usual way to attack that is to study the residuals $\hat{\varepsilon}_t$ that are predictions of the unobservable error terms , to see if they behave like white noise.

To get the residuals I use the arima post-estimation command `predict` with option `residual`:

```
. predict resdy,residual                (This generates resdy = predicted
(1 missing value generated)             error terms (residuals))
```

Do they seem to be Gaussian?



There seems to some mild skewness to the left in relation to a normal pdf. Is this tendency significant? A formal test is, e.g., the Shapiro-Wilk test:

```
. swilk resdy

                   Shapiro-Wilk W test for normal data

    Variable |    Obs       W           V         z       Prob>z
-------------+--------------------------------------------------
       resdy |    141    0.98073      2.126     1.705    0.04413
```

So there is evidence against the normality assumption (5% significance). By reestimating the model using so called robust standard errors, I, to a certain degree, compensate for lack of normality:

```
. arima y, arima(1,1,2) vce(robust)


ARIMA regression

Sample:  1978q2 - 2013q2                        Number of obs      =       141
                                                Wald chi2(3)       =    104.53
Log pseudolikelihood =  456.0505                Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |               Semirobust
         D.y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   .0062495   .0011146     5.61   0.000     .0040649    .0084342
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .7294206   .0940351     7.76   0.000     .5451151     .913726
             |
          ma |
         L1. |  -1.104681   .1138244    -9.71   0.000    -1.327772   -.8815889
         L2. |   .4904761   .0824634     5.95   0.000     .3288508    .6521014
-------------+----------------------------------------------------------------
      /sigma |    .009512   .0005622    16.92   0.000     .0084101    .0106139
------------------------------------------------------------------------------
```

The p-values were not affected, so it does not seem to be too serious this lack of normality.

The next question we should check (which is more important) is: Is there any evidence of autocorrelation left in the residuals?

There are various ways of checking this. The first natural thing to do is to plot the acf and pac (the partial autocorrelation function) which is an integrated part of the Box-Jenkins diagnostic machinery. So I need first to explain the concept of the partial autocorrelation function and its usefulness for time series:

DEF. The partial correlation between two variables, Y_1, Y_2 , controlling for some other variables, $X_1, X_2, \ldots, X_k$, is defined as the correlation between Y_1 and Y_2, in the conditional joint distribution of Y_1, Y_2 given all the

$X_1, X_2, \ldots, X_k$. This correlation can be calculated by drawing out all influence of $X_1, X_2, \ldots, X_k$ on Y_1, Y_2 before the correlation is calculated. I.e., take separately the regression of Y_1 on $X_1, X_2, \ldots, X_k$, and of Y_2 on $X_1, X_2, \ldots, X_k$, take the correlation between the two sets of residuals obtained.

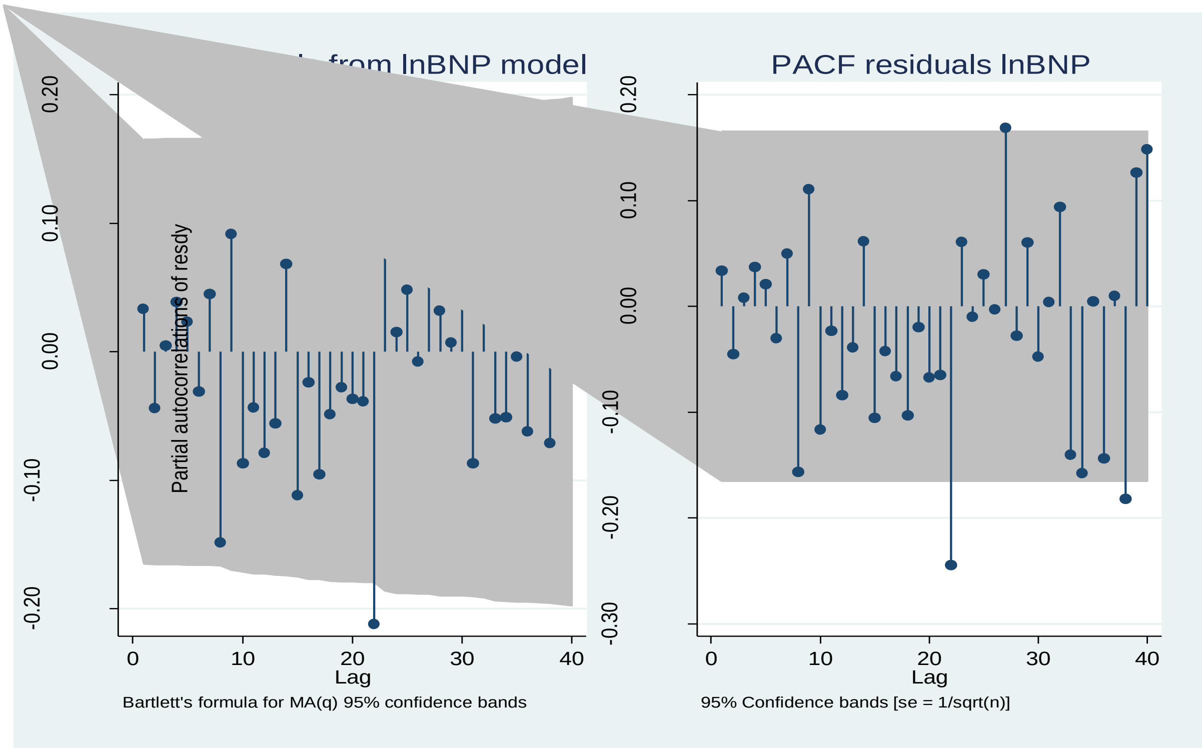
The partial autocorrelation (pac) between Y_t and Y_{t+h} in a stationary time series is the partial correlation controlling for all intermediate Y's, $Y_{t+1}, Y_{t+2}, \ldots, Y_{t+h-1}$. Formulas for pac are not given here but can be found in Hamilton (chap. 3).

The point is, however, that the pair acf(h) and pac(h) can identify a pure AR(p) process and a pure MA(q) process:

For MA(q), acf(h) = 0 for h > q, while pac(h) tends to 0 slower even for h >q.
For AR(p) it is opposite, acf(h) tends to 0 slower and even for h>p, while pac(h) = 0 for h > p.
For an ARMA(p,q) neither acf(h) nor pac(h) get =0 suddenly. They behave similarly.
For white noise both acf(h) and pac(h) are 0 for all h >0.

The asymptotical distribution for the estimated pac(h) have 2 times standard deviation $2/\sqrt{T}$ that we use as a benchmark for random variation in the estimated pac when the true pac(h) = 0.

The commands `ac resdy` and `pac resdy` give plots:



The grey area represents the natural variation (95%) for the estimated acf and pac when the true acf and pac are 0.

Except for possibly one *h*, the plots does not seem provide any evidence against the two functions of *h* being zero.

The command `corrgram` gives more info on this with *the Portmanteau test Q* for testing the white noise null hypothesis, H_0, that acf(h) = 0 for all $0 < h \leq m$.

$$Q_m = T(T+2)\sum_{j=1}^{m} \frac{1}{T-j}\hat{\rho}^2(j)$$

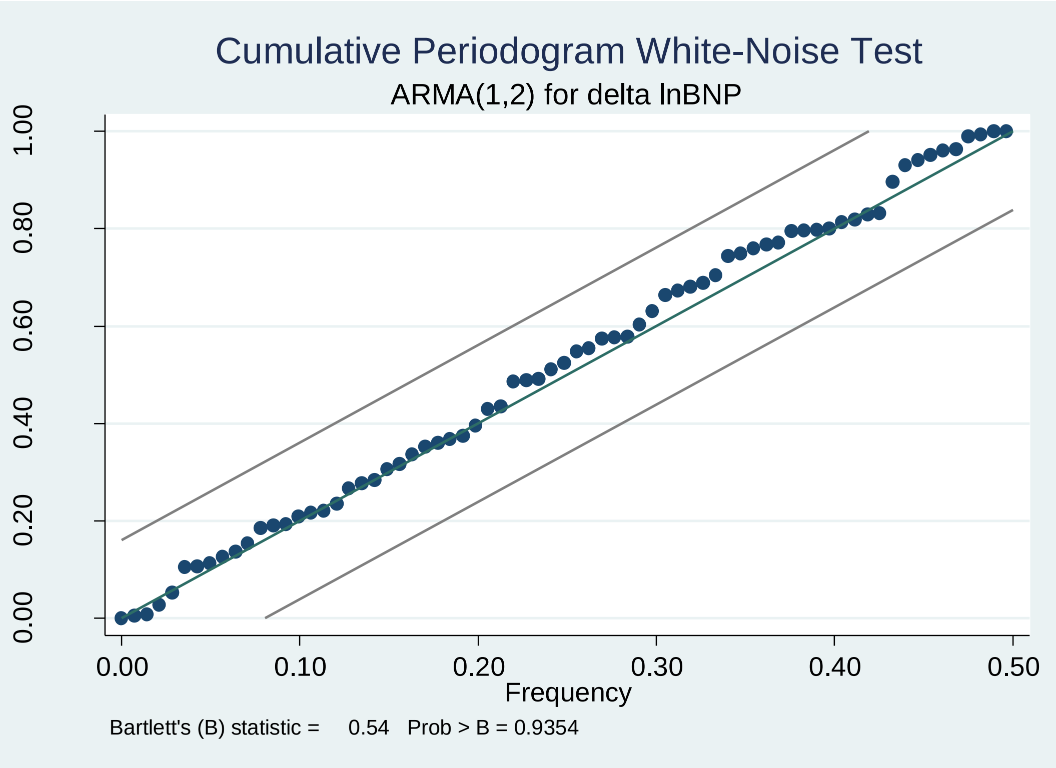
Under H_0, it can be proven that $Q_m \underset{T\to\infty}{\overset{F}{\longrightarrow}} \chi^2_m$ (chi-square distr. with *m* degrees of freedom. Reject H_0 for large values of Q_m.

```
. corrgram resdy, lags(25)

                                          -1       0       1 -1       0       1
 LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1        0.0337   0.0337   .16313  0.6863          |                  |
2       -0.0436  -0.0449   .43906  0.8029          |                  |
3        0.0051   0.0081   .44284  0.9313          |                  |
4        0.0387   0.0371   .66354  0.9557          |                  |
5        0.0237   0.0210   .74648  0.9803          |                  |
6       -0.0307  -0.0299   .88739  0.9895          |                  |
7        0.0451   0.0499   1.1931  0.9911          |                  |
8       -0.1480  -0.1563   4.5145  0.8080         -|                 -|
9        0.0917   0.1107   5.7994  0.7598          |                  |
10      -0.0869  -0.1160   6.9607  0.7292          |                  |
11      -0.0434  -0.0232   7.2523  0.7783          |                  |
12      -0.0786  -0.0837   8.2185  0.7678          |                  |
13      -0.0558  -0.0385   8.7095  0.7945          |                  |
14       0.0684   0.0616   9.4528  0.8010          |                  |
15      -0.1117  -0.1050   11.449  0.7201          |                  |
16      -0.0239  -0.0422   11.542  0.7749          |                  |
17      -0.0954  -0.0658   13.021  0.7348          |                  |
18      -0.0485  -0.1030   13.407  0.7668          |                  |
19      -0.0276  -0.0198   13.533  0.8102          |                  |
20      -0.0365  -0.0669   13.755  0.8427          |                  |
21      -0.0386  -0.0648   14.005  0.8694          |                  |
22      -0.2117  -0.2445   21.601  0.4839         -|                 -|
23       0.1105   0.0612   23.688  0.4212          |                  |
24       0.0152  -0.0095   23.728  0.4772          |                  |
25       0.0486   0.0301   24.138  0.5114          |                  |
```

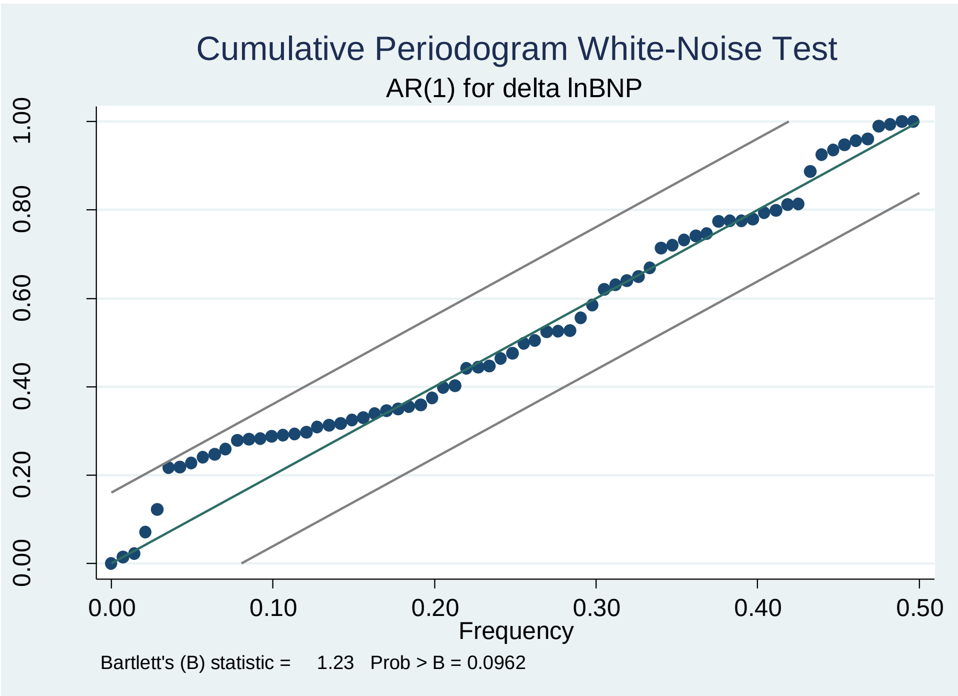
We see no no evidence against white noise her even when the suspicious values at lags 8 and 22 are included in *Q*.

There are other tests as well (e.g., the runs test for independence – not shown here) and the Bartlett test shown below:



Under the white noise hypothesis the residuals should lie along the straight line, which seems to be the case for this model. The p-value about 0.9.

In contrast I show the same test for another tentative model, i.e., the AR(1):

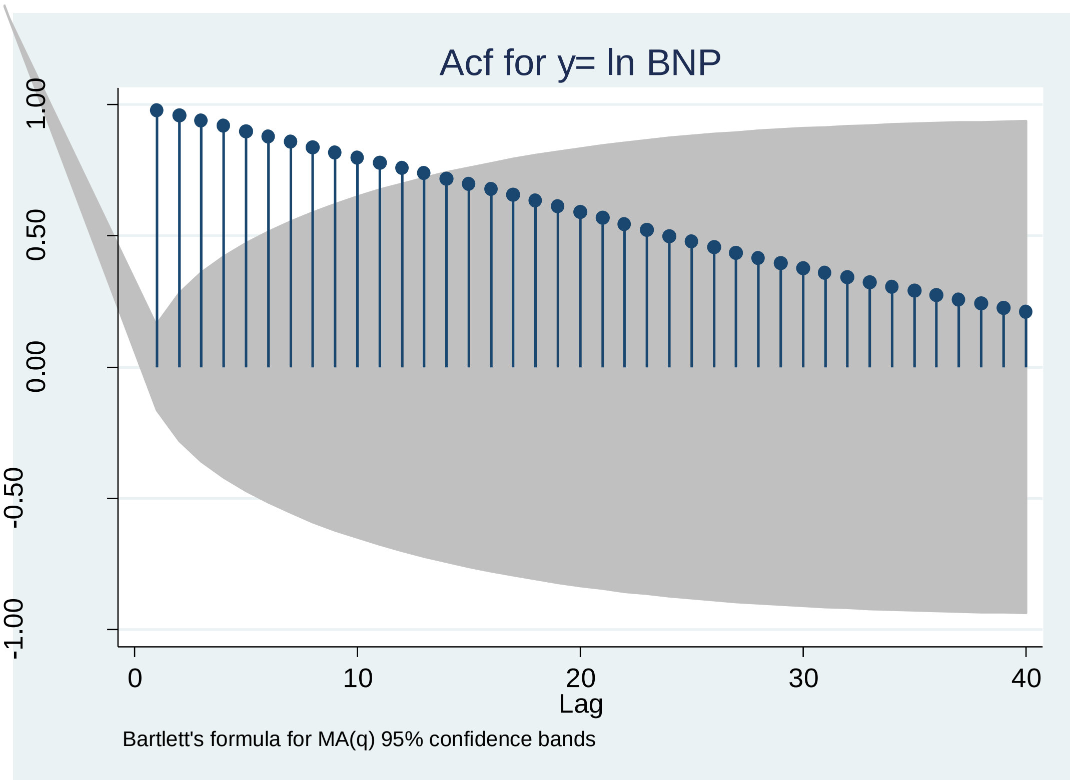


5 Identification

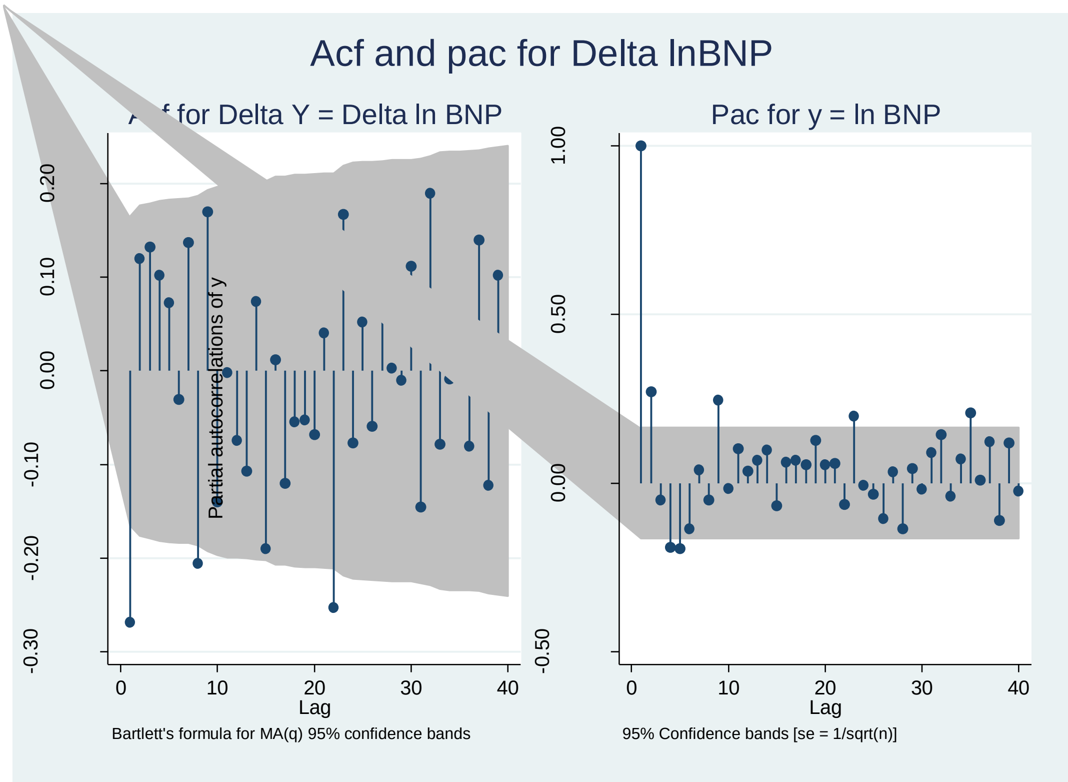
So how did I arrive at the conclusion that the ARMA(1,2) is the “best” ARMA model for $\Delta \ln BNP$?

The apparatus for the Box-Jenkins identification procedure is more or less established except for one point I am coming to – the use of information criteria.

First I look at the acf for the original series $Y_t = \ln BNP_t$:



This is a typical acf.graph for a non-stationary process – a causal ARMA should have an acf that tends to zero in an exponential manner. This is clearly not the case here – indicating differencing as the first step.



The acf/pac graph does not show any clear preference for a pure AR or pure MA, so I will try several ARMA(p,q).

My strategy is to try all 25 ARMA(p,q) modeler med $0 \le p, q \le 4$.
It was early recognized that it is not a good idea to compare models just by looking a the max log likelihood value, since this automatically increases when new parameters are added to the model. It turns out that over-fitting (too many parameters) in a model seriously increases uncertainty in particular in connection with prediction which is the most common use of arima models. So criteria have been developed that combine the information in the log likelihood with a punishment due to number of parameters. Two criteria have turned out to work well in time series contexts – i.e., the Akaike information criterion (AIC) and the Schwartz Bayesian information criterion (B):

AIC: Choose a model to minimize $AIC = -2 \times \ln(\text{likelihood}) + 2k$

BIC: Choose a model to minimize $BIC = -2 \times \ln(\text{likelihood}) + 2\ln(T) \times k$

`where

$T =$ number of observations

$k =$ number of parameters

Simulations show that AIC sometimes lead to larger than necessary models. However, from other principles both criteria are well founded. There are also other criteria suggested in the literature, but the two mentioned are the most common.

In Stata the arima post-estimation command `estat ic` provide both. The following table show the AIC and the BIC criteria for the 25 models:

AIC	MA				
AR order	0	1	2	3	4
0	-882.7	-889.1	-893.1	-897.6	-899.1
1	-891.4	-889.5	-902.1	-901.0	-899.1
2	-889.7	-893.3	-892.9	-904.6	-904.0
3	-893.0	-896.2	-899.5	-902.8	-900.9
4	-896.5	-896.0	-897.5	-902.0	-894.5
5	-897.0				
6	-895.0				
BIC	MA order				
AR order	0	1	2	3	4
0	-876.8	-880.3	-881.3	-882.9	-881.4
1	-882.5	-877.7	-887.4	-883.3	-878.5
2	-877.9	-878.6	-875.2	-883.9	-883.4
3	-878.3	-878.5	-881.9	-879.2	-874.4
4	-878.8	-875.3	-873.9	-878.4	-865.0
5	-876.4				
6	-871.7				

The "winner" model for AIC is ARMA(2,3), and the "winner" for BIC is ARMA(1,2). Using the principle that, when, forecasting is the main purpose of the model, then then the fewer parameters, the better. So I went for the ARMA(1,2) and subjected it to some specification testing as done above to decide if it is acceptable.

It passed the tests, so I decided ARMA(1,2) as my model for forecasting.

6. Forecasting.

Here I just show some example forcasting graphs produced by Stata for the BNP data and leave the theoretical justification to next lecture.

The arima post-estimation commands give many forcasting possibilities. In order to enable the arima post-estimation commands, you have run the arima estimation command first. Which I did showing the output above.

```
. arima y,arima(1,1,2)            (I have defined y = ln(bnp) in my dataset)
```

(---- Output shown on page 11 ----)

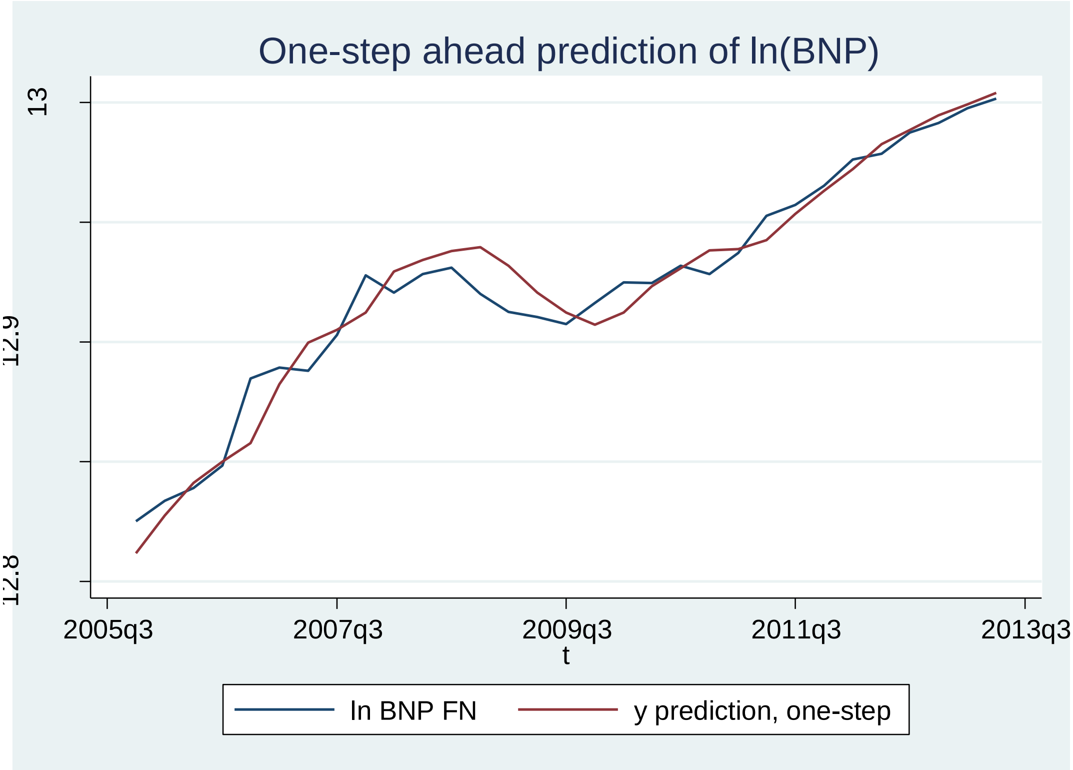
```
. predict resdy,residual                (This generates resdy = predicted
(1 missing value generated)             error terms (residuals))

. predict preddy,xb               (The option xb generates preddy =
                                        (one-step ahead) predicted delta(lnBNP))

. predict yhat,y                  (The option y generates yhat =
(1 missing value generated)             (one-step ahead) predicted y=lnBNP )

. list y yhat D.y preddy resdy in 1/20
```

	y=lnBNP	yhat	D.y= delta y	preddy	residual resdy
1.	12.1191	.	.	.0062495	.
2.	12.1404	12.12534	.021321	.0062495	.0150718
3.	12.1405	12.14201	.000117	.0015945	-.0014772
4.	12.1448	12.14976	.004294	.0092284	-.004934
5.	12.162	12.15409	.01715	.0092649	.0078851
6.	12.1631	12.16553	.001154	.0035547	-.0024007
7.	12.1633	12.17194	.000182	.008814	-.0086319
8.	12.1711	12.1735	.007751	.0101855	-.002435
9.	12.1937	12.17688	.02261	.0058132	.0167965
10.	12.1808	12.19214	-.012824	-.0015284	-.0112957
11.	12.1955	12.19387	.01464	.0130202	.0016197
12.	12.2048	12.20054	.00928	.0050488	.0042314
13.	12.2072	12.20935	.00246	.0045802	-.0021207
14.	12.2002	12.21513	-.007067	.0079026	-.0149694
15.	12.1977	12.21219	-.002482	.0120319	-.0145143
16.	12.2008	12.20625	.003141	.0085723	-.0054309
17.	12.2106	12.20368	.009733	.0028631	.0068701
18.	12.2064	12.20909	-.00418	-.0014623	-.0027176
19.	12.2037	12.21139	-.002623	.0050138	-.0076364
20.	12.2108	12.21063	.007057	.0068808	.0001764



7. Dynamic forcasting in Stata.

You need dynamic forcasting if you wish to forcast more than one period ahead. So, first I use only data up to 2010q1 to estimate my model and predict the rest of the period from these. Then I can compare these predictions with the actual observations:

```
. arima y if tin(,2010q1), arima(1,1,2)


ARIMA regression

Sample:  1978q2 - 2010q1                        Number of obs      =       128
                                                Wald chi2(3)       =     82.63
Log likelihood =  409.2349                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
         D.y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   .0062803   .0013115     4.79   0.000     .0037098    .0088509
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .7118857   .1351215     5.27   0.000     .4470524    .9767189
             |
          ma |
         L1. |  -1.096477   .1254912    -8.74   0.000    -1.342436   -.8505192
         L2. |   .5054851   .0900681     5.61   0.000     .3289549    .6820153
-------------+----------------------------------------------------------------
      /sigma |   .0098696   .0007075    13.95   0.000      .008483    .0112562
------------------------------------------------------------------------------
```

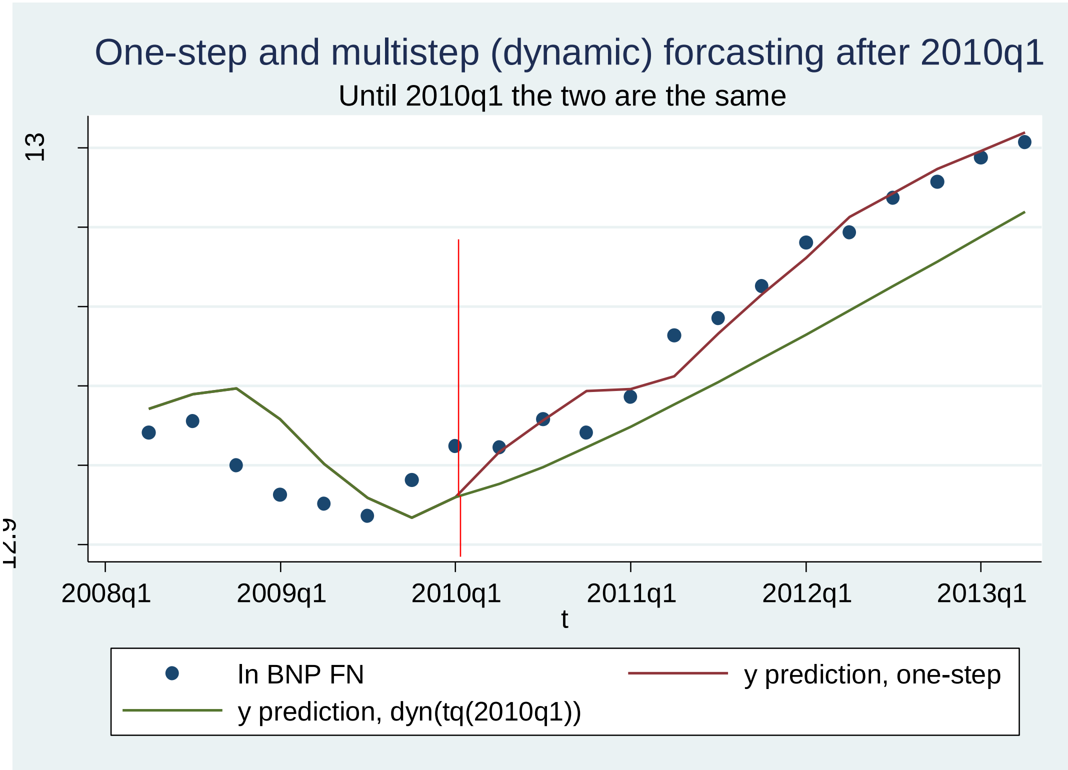
```
. predict shorty,y            (This generates one-step ahead predictions for the
(1 missing value generated)     whole period until 2013q1


. predict shortdy, dynamic(tq(2010q1)) y
(1 missing value generated)

. twoway (tsline y if t>tq(2008q1), recast(scatter)) (tsline shorty if t>tq(2008q1))
            (tsline shortdy if t>tq(2008q1))
```

From the Stata pdf-documentation on arima postestimation:

> `dynamic(tq(2010q1))` produces dynamic (also known as recursive) forecasts. time constant specifies when the forecast is to switch from one step ahead to dynamic. In dynamic forecasts, references to y_t evaluate to the prediction of y_t for all periods at or after time constant; they evaluate to the actual value of y_t for all prior periods.



8 Redundant parameters (the common factor problem)

In the next example "eps" is 142 observations of white noise $\varepsilon_t \sim N(0,1)$. Fitting an ARMA(1,1) gives 2 (spuriously) significant parameters.

Rule: Cancel common factors in $\varphi(L)$ and $\theta(L)$ in an ARMA-model in order to avoid spuriously significant parameters. So, when we talk about a proper ARMA model we implicitly assume that $\varphi(L)$ and $\theta(L)$ have no common factors.

```
. summarize eps

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         eps |       142     .010147    1.022428  -2.885089   1.837664


arima eps, arima(1,0,1)

ARIMA regression

Sample:  1978q1 - 2013q2                        Number of obs      =       142
                                                Wald chi2(2)       =     22.46
Log likelihood = -203.8354                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
         eps |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
eps          |
       _cons |   .0100242   .0876471     0.11   0.909    -.1617609    .1818093
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |  -.9100905   .2608814    -3.49   0.000    -1.421409   -.3987724
             |
          ma |
         L1. |   .8835049   .2927291     3.02   0.003     .3097663    1.457243
-------------+----------------------------------------------------------------
      /sigma |   1.016568   .0713344    14.25   0.000      .876755    1.156381
```

Note that in this model we know that (*) $Y_t = \varepsilon_t \sim WN(0,1)$. This implies that for any constant, c, we have (**) $cY_{t-1} = c\varepsilon_{t-1}$,
Subtracting (**) from (*), we get $(1-cL)Y_t = (1-cL)\varepsilon_t$, so that the ARMA(1,1) is true for any c. By running the ARIMA(1,0,1) we see that we get significant values for φ_1 og $-\theta_1$, which clearly would be misleading if we did not know that Y_t were white noise.

Hence, in order that an ARMA(p,q) model does not give misleading results, we should require that the two polynomials, $\varphi(L)$ and $\theta(L)$ doe not have any common factors. In other words:

Cancel all common factors in the ratio $\dfrac{\theta(L)}{\varphi(L)}$*, in order to have a proper ARMA-model!*