

時系列分析

はじめに

- **arima** コマンドは、被説明変数で自己回帰する標準の ARIMA モデルと、ARMA 誤差項を伴う構造モデルの両方にフィットを行います。
- 一次自己回帰移動平均(Autoregressive Moving Average)過程において、**arima** はモデルの全てのパラメータを次のように推定します。

$$y_t = x_t\beta + \mu_t$$

構造方程式

$$\mu_t = \rho\mu_{t-1} + \theta\epsilon_{t-1} + \epsilon_t$$

攪乱、ARMA(1,1)

ρ	一次自己相関パラメータ
θ	一次移動平均パラメータ
ϵ_t	$\sim i.i.d.N(0, \sigma^2)$ 、 ϵ_t はホワイトノイズ攪乱

上記2つの式は組み合わせることができ、攪乱過程の一般的な ARIMA(p, q)は次のように記述します。

$$y_t = x_t\beta + \rho_1(y_{t-1} - x_{t-1}\beta) + \rho_2(y_{t-2} - x_{t-2}\beta) + \dots + \rho_p(y_{t-p} - x_{t-p}\beta) \\ + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \dots + \theta_q\epsilon_{t-q} + \epsilon_t$$

次のようにラグ演算子表記を使用して、ARMA モデルの一般的な形式をより簡潔に記述することもできます。

$$\rho(L^p)(y_t - x_t\beta) = \theta(L^q)\epsilon_t \quad \text{ARMA}(p, q)$$

ここで

$$\rho(L^p) = 1 - \rho_1L - \rho_2L^2 - \dots - \rho_pL^p$$

$$\theta(L^q) = 1 + \theta_1L + \theta_2L^2 + \dots + \theta_qL^q$$

$$L^j y_t = y_{t-j}$$

乗算の季節的 ARMAX および ARIMA モデルもフィットできます。

- 定常系列の場合、完全または無条件の最尤推定値はカルマンフィルタを介して取得されます。
- 定常系列でない場合で、ある程度情報が揃っていれば、`state0()` コマンドや `p0()` コマンドでフィルタの初期値の指定や、`diffuse` オプションを使用して無情報事前分布を仮定できます。

ARIMA モデル

- 構造コンポーネントのない純粋な ARIMA モデルにはリグレッサがなく、構造方程式からの攪乱項の自己回帰ではなく、被説明変数の自己回帰として記述されます。例として、ARIMA(1, 1)モデルは次のように記述されます。

$$y_t = \alpha + \rho y_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \quad (1a)$$

- これらのモデルは `arima` によって推定された攪乱項定式化の ARMA と同等ですが（定数項 α のスケールファクターを除く）、後者はより柔軟で、より幅広いクラスのモデルを可能にします。この効果を確認するために、上記の構造方程式の $x_t \beta$ を定数項 β_0 に置き換えます。

$$\begin{aligned} y_t &= \beta_0 + \mu_t \\ &= \beta_0 + \rho \mu_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \\ &= \beta_0 + \rho(y_{t-1} - \beta_0) + \theta \epsilon_{t-1} + \epsilon_t \\ &= (1 - \rho)\beta_0 + \rho y_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \end{aligned} \quad (1b)$$

- 式(1a)と式(1b)は $\alpha = (1 - \rho)\beta_0$ で同等なので、ARIMA モデルを被説明変数の自己回帰または攪乱のどちらと見なすかどうかは重要ではありません。この解釈は、ARMA(1, 1)ケースから一般的な ARIMA(ρ, d, q)ケースに簡単に拡張できます。

例題 1：ARIMA モデル

- Enders(2004, 87-93)は、米国卸売物価指数 (WPI) の 1960 年第一四半期から 1990 年第四四半期のデータを使用して、ARIMA モデルを検討しています。
- 階差と自己回帰成分と移動平均成分の両方を含む最も単純な ARIMA モデルは ARMA(1, 1, 1)です。

データセットをインポートしてフィットを実行します。Stata のコマンドウィンドウで次のコマンドを実行します。

```
use https://www.stata-press.com/data/r16/wpi1  
  
arima wpi, arima(1,1,1)
```

```
(setting optimization to BHHH)
Iteration 0: log likelihood = -139.80133
Iteration 1: log likelihood = -135.6278
Iteration 2: log likelihood = -135.41838
Iteration 3: log likelihood = -135.36691
Iteration 4: log likelihood = -135.35892
(switching optimization to BFGS)
Iteration 5: log likelihood = -135.35471
Iteration 6: log likelihood = -135.35135
Iteration 7: log likelihood = -135.35132
Iteration 8: log likelihood = -135.35131
```

ARIMA regression

```
Sample: 1960q2 - 1990q4                Number of obs   =      123
                                         Wald chi2(2)    =     310.64
Log likelihood = -135.3513              Prob > chi2     =     0.0000
```

D.wpi		OPG		z	P> z	[95% Conf. Interval]	
	Coef.	Std. Err.					
wpi							
	_cons	.7498197	.3340968	2.24	0.025	.0950019	1.404637
ARMA							
	ar						
	L1.	.8742288	.0545435	16.03	0.000	.7673256	.981132
	ma						
	L1.	-.4120458	.1000284	-4.12	0.000	-.6080979	-.2159938
	/sigma	.7250436	.0368065	19.70	0.000	.6529042	.7971829

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

- 推定結果から、AR(1)の係数は 0.874、MA(1)の係数は-0.412 で、どちらも有意です。
- 推定されたホワイトノイズ攪乱 ϵ は 0.725 です。

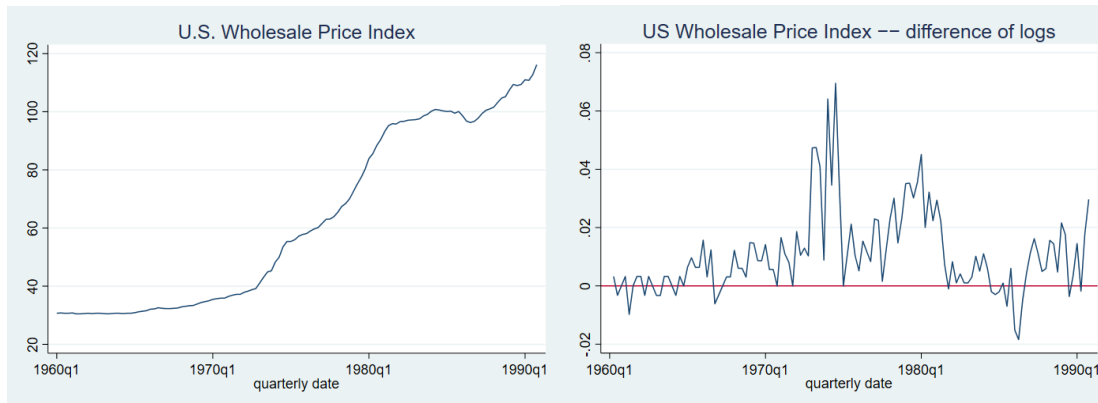
上記のモデルは次のように記述することもできます。

```
arima D.wpi, ar(1) ma(1)
```

- 説明変数 **wpi** の前にある **D.** は、階差を示す Stata の時系列演算子です。
- 1960 年第二四半期から 1990 年第四四半期の米国卸売物価指数 (WPI) の第一階差をモデル化します。階差のために最初の観測値が失われるためです。

例題 2：付加的な季節効果を持つ ARIMA モデル

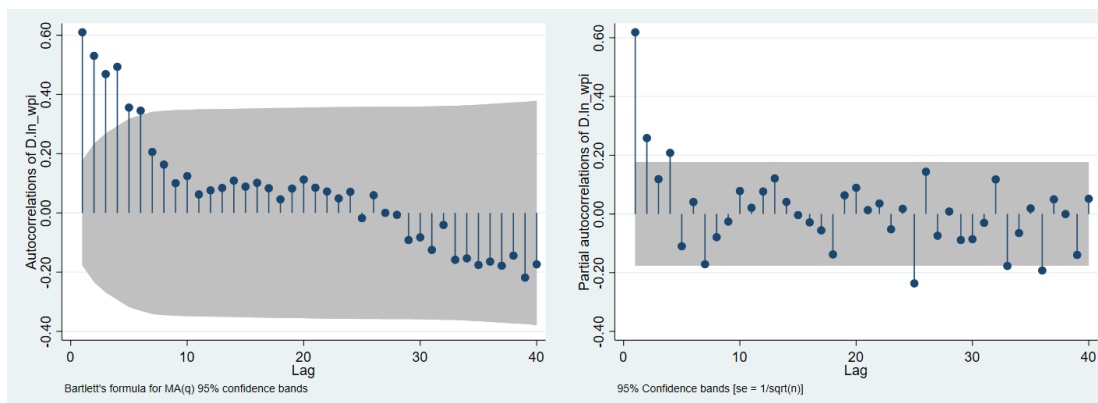
- Enders は階差のある系列の分散を安定させるために、自然対数の階差のモデルを選択しました。原系列と対数の第一階差をグラフ化すると次のようになります。



- Enders は自己相関、偏自己相関（下図参照）および例題 1 の推定結果に基づき、対数階差系列の ARMA モデルを作成しました。

```
ac D.ln_wpi, ylabel(-.4(.2).6)
```

```
pac D.ln_wpi, ylabel(-.4(.2).6)
```



- 自己回帰項と MA(1)項に加えて、残りの四半期効果を説明するために MA(4)項が含まれています。モデルは次のようにフィットできます。

$$\Delta \ln(wpi_t) = \beta_0 + \rho_1 \{\Delta \ln(wpi_{t-1}) - \beta_0\} + \theta_1 \epsilon_{t-1} + \theta_4 \epsilon_{t-4} + \epsilon_t$$

このモデルを `arima` と Stata の階差演算子を用いてフィットします。

```
arima D.ln_wpi, ar(1) ma(1 4)
```

```
(setting optimization to BHHH)
Iteration 0:  log likelihood = 382.67447
Iteration 1:  log likelihood = 384.80754
Iteration 2:  log likelihood = 384.84749
Iteration 3:  log likelihood = 385.39213
Iteration 4:  log likelihood = 385.40983
(switching optimization to BFGS)
Iteration 5:  log likelihood = 385.9021
Iteration 6:  log likelihood = 385.95646
Iteration 7:  log likelihood = 386.02979
Iteration 8:  log likelihood = 386.03326
Iteration 9:  log likelihood = 386.03354
Iteration 10: log likelihood = 386.03357
```

ARIMA regression

```
Sample: 1960q2 - 1990q4                Number of obs   =      123
                                         Wald chi2(3)    =     333.60
Log likelihood = 386.0336                Prob > chi2     =     0.0000
```

D.ln_wpi	Coef.	OPG Std. Err.	z	P> z	[95% Conf. Interval]	
ln_wpi						
_cons	.0110493	.0048349	2.29	0.022	.0015731	.0205255
ARMA						
ar						
L1.	.7806991	.0944946	8.26	0.000	.5954931	.965905
ma						
L1.	-.3990039	.1258753	-3.17	0.002	-.6457149	-.1522928
L4.	.3090813	.1200945	2.57	0.010	.0737003	.5444622
/sigma	.0104394	.0004702	22.20	0.000	.0095178	.0113609

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

- 結果から、対数階差系列は 0.781 のレベルで強い自己相関を示しています。
- イノベーションは次の四半期にマイナス 0.399 の影響を与え、翌年にはプラス 0.309 の季節的影響を与えます。

例題 3：乗算 SARIMA モデル

- 最も一般的な乗算 SARIMA モデルは Box, Jenkins, and Reinsel (2008, sec. 9.2)の $(0, 1, 1) \times (0, 1, 1)_{12}$ の「エアライン」モデルです。
- データセット **airline.dta** には、1949 年 1 月から 1960 年 12 月までの毎月の国際航空旅客データが含まれています。
- データの第一階差、季節階差を取ります。トレンドコンポーネントの存在を疑わないため、**arima** で **noconstant** オプションを使用します。

データセットをインポートしてフィットを実行します。

```
use https://www.stata-press.com/data/r16/air2

generate lnair = ln(air)

arima lnair, arima(0,1,1) sarima(0,1,1,12) noconstant
```

```
(setting optimization to BHHH)
Iteration 0:  log likelihood = 223.8437
Iteration 1:  log likelihood = 239.80405
Iteration 2:  log likelihood = 244.10265
Iteration 3:  log likelihood = 244.65895
Iteration 4:  log likelihood = 244.68945
(switching optimization to BFGS)
Iteration 5:  log likelihood = 244.69431
Iteration 6:  log likelihood = 244.69647
Iteration 7:  log likelihood = 244.69651
Iteration 8:  log likelihood = 244.69651
```

ARIMA regression

```
Sample: 14 - 144                                Number of obs   =      131
                                                Wald chi2(2)    =      84.53
Log likelihood = 244.6965                       Prob > chi2     =      0.0000
```

DS12.lnair	OPG					[95% Conf. Interval]
	Coef.	Std. Err.	z	P> z		
ARMA						
ma						
L1.	-.4018324	.0730307	-5.50	0.000	-.5449698	-.2586949
ARMA12						
ma						
L1.	-.5569342	.0963129	-5.78	0.000	-.745704	-.3681644
/sigma	.0367167	.0020132	18.24	0.000	.0327708	.0406625

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

- 結果から、毎月の国際航空旅客数のモデルは次のようになります。

$$\Delta\Delta_{12}\lnair_t = -0.402\epsilon_{t-1} - 0.557\epsilon_{t-12} + 0.224\epsilon_{t-13} + \epsilon_t$$

$$\hat{\sigma} = 0.037$$

- 例えば ln(2) の場合、 ϵ_{t-13} の係数は ϵ_{t-1} および ϵ_{t-12} の項の係数の積です ($0.224 \approx -0.402 \times -0.557$)。

- `arima` では、被説明変数に `DS12.lnair` のようにラベルを付けて、階差演算子 Δ とラグ 12 季節階差演算子 Δ_{12} を適用できます。

次のように記述します。この構文ではより複雑な季節項を組み込むことができます。

```
arima DS12.lnair, ma(1) mma(1, 12) noconstant
```

例題 4：ARMAX モデル

- 共変量を含むモデルの簡単な例については、貨幣数量説を表す Friedman and Meiselman(1963)の方程式の更新版を使用します。個人消費支出(`consump`)と M2 (`m2`) で測定されるマネーサプライとの間の直接的な関係を仮定します。

$$\text{consump}_t = \beta_0 + \beta_1 m2_t + \mu_t$$

- このモデルを 1959 年第一四半期から 1981 年第四四半期のデータでフィットします。1982 年以前の期間に限定する理由は、連邦準備制度が 1980 年代後半にインフレを制御するためにマネーサプライを広範囲に操作し、消費とマネーサプライの関係が複雑化したためです。
- モデルには、攪乱の自己回帰項と移動平均項の両方を含めます。構造方程式の攪乱を次のようにモデル化します。この関係性は季節調整済みデータで推定されているため、季節効果を明確に含める必要はありません。

$$\mu_t = \rho\mu_{t-1} + \theta\epsilon_{t-1} + \epsilon_t$$

- `if` 式で `tin()` 関数を使用して、推定を目的のサンプルに制限します。`tin()` の最初の引数を空白のままにすることで、2 番目の日付 (1981 年第四四半期) までに利用可能なすべてのデータを含めます。

データセットをインポートしてフィットを実行します。

```
use https://www.stata-press.com/data/r16/friedman2, clear

arima consump m2 if tin(, 1981q4), ar(1) ma(1)
```

```
(setting optimization to BHHH)
Iteration 0: log likelihood = -344.67575
Iteration 1: log likelihood = -341.57248
Iteration 2: log likelihood = -340.67391
Iteration 3: log likelihood = -340.57229
Iteration 4: log likelihood = -340.5608
(switching optimization to BFGS)
Iteration 5: log likelihood = -340.5515
Iteration 6: log likelihood = -340.51272
Iteration 7: log likelihood = -340.50949
Iteration 8: log likelihood = -340.5079
Iteration 9: log likelihood = -340.50775
Iteration 10: log likelihood = -340.50774
```

ARIMA regression

```
Sample: 1959q1 - 1981q4                Number of obs   =      92
                                         Wald chi2(3)    =    4394.80
Log likelihood = -340.5077              Prob > chi2     =     0.0000
```

		OPG			[95% Conf. Interval]	
consump		Coef.	Std. Err.	z	P> z	
<hr/>						
consump						
	m2	1.122029	.0363563	30.86	0.000	1.050772 1.193286
	_cons	-36.09872	56.56703	-0.64	0.523	-146.9681 74.77062
<hr/>						
ARMA						
	ar					
	L1.	.9348486	.0411323	22.73	0.000	.8542308 1.015467
	ma					
	L1.	.3090592	.0885883	3.49	0.000	.1354293 .4826891
<hr/>						
	/sigma	9.655308	.5635157	17.13	0.000	8.550837 10.75978

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

- 消費は所得速度の1つの側面にすぎませんが、この期間の消費(1.122)に関しては比較的小さな貨幣の流通速度が見られます。
- 攪乱項における非常に大きな一次自己相関、および統計的に有意な一次移動平均にも注目します。

このモデルで、誤差項の不均一分散や非正規分布につながっている心配があるときは、`vce(robust)` オプションを使用してモデルを再度フィットします。

```
arima consump m2 if tin(, 1981q4), ar(1) ma(1) vce(robust)
```

```
(setting optimization to BHHH)
Iteration 0: log pseudolikelihood = -344.67575
Iteration 1: log pseudolikelihood = -341.57248
Iteration 2: log pseudolikelihood = -340.67391
Iteration 3: log pseudolikelihood = -340.57229
Iteration 4: log pseudolikelihood = -340.5608
(switching optimization to BFGS)
Iteration 5: log pseudolikelihood = -340.5515
Iteration 6: log pseudolikelihood = -340.51272
Iteration 7: log pseudolikelihood = -340.50949
Iteration 8: log pseudolikelihood = -340.5079
Iteration 9: log pseudolikelihood = -340.50775
Iteration 10: log pseudolikelihood = -340.50774
```

ARIMA regression

```
Sample: 1959q1 - 1981q4                Number of obs   =      92
                                         Wald chi2(3)    =    1176.26
Log pseudolikelihood = -340.5077       Prob > chi2     =     0.0000
```

		Semirobust		z	P> z	[95% Conf. Interval]	
consump		Coef.	Std. Err.				
consump							
	m2	1.122029	.0433302	25.89	0.000	1.037103	1.206954
	_cons	-36.09872	28.10478	-1.28	0.199	-91.18308	18.98564
ARMA							
	ar						
	L1.	.9348486	.0493428	18.95	0.000	.8381385	1.031559
	ma						
	L1.	.3090592	.1605359	1.93	0.054	-.0055854	.6237038
/sigma		9.655308	1.082639	8.92	0.000	7.533375	11.77724

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

- 推定標準誤差が大幅に増加していることがわかります。また、初回のフィットでは明らかに有意であった移動平均項が、今回のフィットではわずかに有意となっていることがわかります。

arima 事後推定

例題 1：ダイナミック予測

- **arima** コマンドの魅力は、ダイナミックな予測を行う機能です。ARIMA モデルの例題 4 では次のモデルをフィットしました。

$$\begin{aligned} \text{consump}_t &= \beta_0 + \beta_1 m2_t + \mu_t \\ \mu_t &= \rho \mu_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \end{aligned}$$

- はじめに 1978 年の第一四半期までのデータを使用してモデルを再度フィットし、次に 1 期先のダイナミックな予測を評価します。

データセットをインポートしてフィットを実行します。予測については、被説明変数には時系列演算子が含まれていないため、代わりに **predict chat, xb** を使用します。

```
use https://www.stata-press.com/data/r16/friedman2

keep if time<=tq(1981q4)

arima consump m2 if tin(, 1978q1), ar(1) ma(1)

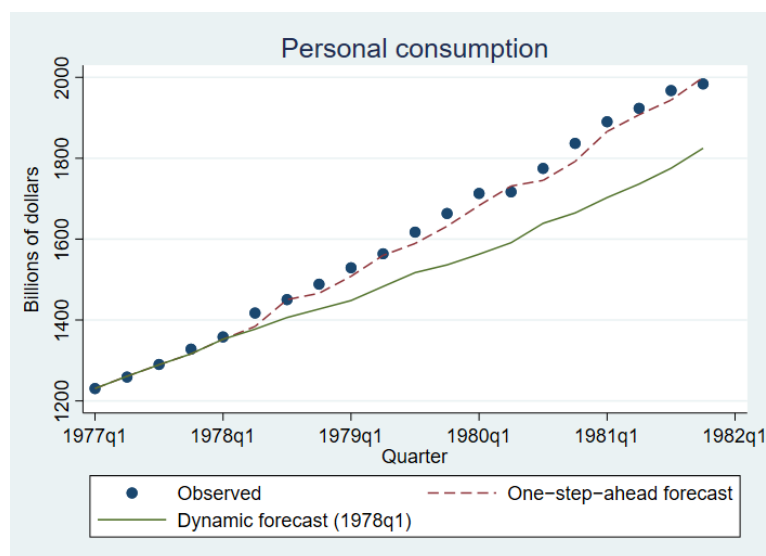
predict chat, y
```

(結果は省略)

- 1978 年第一四半期の消費の観測値から予測値に切り替えて、ダイナミックな予測を行います。

```
predict chatdy, dynamic(tq(1978q1)) y
```

- 次のグラフは、最初の数年間の観測値と予測値を比較しています。

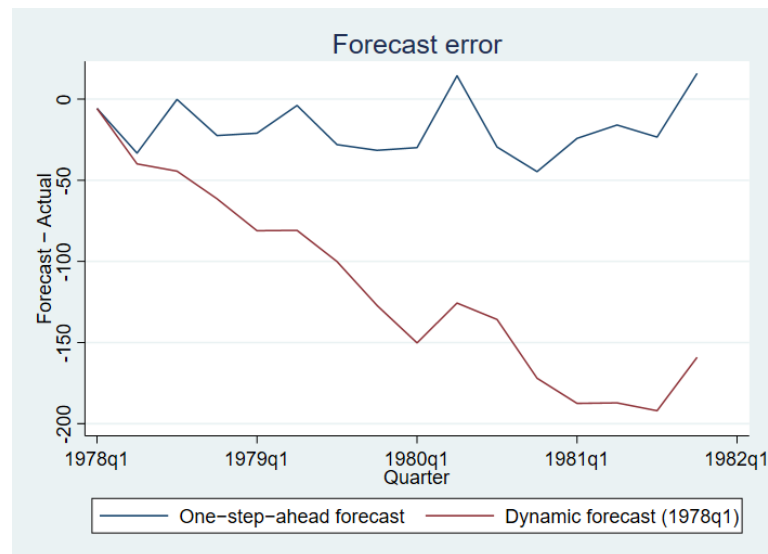


- 1期先予測は、時間の経過とともに観測値から大きく外れることはありません。
- 一方、ダイナミック予測は時間の経過とともに誤差が大きくなります。この原因を探るために、モデルを次のように書き換えます。

$$\begin{aligned} \text{consump}_t &= \beta_0 + \beta_1 m2_t + \rho \mu_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \\ &= \beta_0 + \beta_1 m2_t + \rho(\text{consump}_{t-1} - \beta_0 - \beta_1 m2_{t-1}) + \theta \epsilon_{t-1} + \epsilon_t \end{aligned}$$

- この式は、時間 t での消費の予測値が時間 $t-1$ での消費の値に依存することを示しています。期間 t の1期先予測を行うとき、時間 $t-1$ での消費の実際の値がわかっています。
- 一方で、`dynamic(tq(1978q1))` オプションを使用すると次のようになります。
 - 1978年第一四半期の予測値 ← 1977年第四四半期の観測値に基づく
 - 1978年第二四半期の予測値 ← 1978年第一四半期の予測値に基づく
 - 1978年第三四半期の予測値 ← 1978年第二四半期の予測値に基づく
 - ...

したがって、ダイナミック予測では、以前の予測誤差は時間の経過とともに蓄積されます。次のグラフはこの効果を示しています。



例題 2 : ARIMA の IRF (インパルス応答関数) 結果

- Enders (2004)の M1 で測定された、四半期ごとの米国のマネーサプライのモデルを考えてみましょう。
- Enders (2004, 93-97)は、季節的な購買パターンが M1 で季節的影響を引き起こす理由について説明しています。変数 `lnm1` は、マネーサプライの自然対数データです。
- 季節性、非季節性 ARIMA モデルでフィットを実行し、各モデルで計算された IRF を比較します。
- 次の非季節性 ARIMA モデルを使用します。

$$\Delta\Delta_4 \ln m1_t = \rho_1(\Delta\Delta_4 \ln m1_{t-1}) + \rho_4(\Delta\Delta_4 \ln m1_{t-4}) + \epsilon_t$$

データセットをインポートしてフィットを実行します。

```
use https://www.stata-press.com/data/r16/m1nsa, clear
arima DS4.lnm1, ar(1 4) noconstant nolog
```

ARIMA regression

Sample: 1961q2 - 2008q2

Number of obs = 189

Wald chi2(2) = 78.34

Log likelihood = 579.3036

Prob > chi2 = 0.0000

DS4.lnm1	Coef.	OPG Std. Err.	z	P> z	[95% Conf. Interval]	
ARMA						
ar						
L1.	.3551862	.0503011	7.06	0.000	.2565979	.4537745
L4.	-.3275808	.0594953	-5.51	0.000	-.4441895	-.210972
/sigma	.0112678	.0004882	23.08	0.000	.0103109	.0122246

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

次のコマンドを実行して、IRF 結果を `myirf.irf` というファイルに保存します。

```
irf create nonseasonal, set(myirf) step(30)
```

- 次に季節性 ARIMA モデルを使用します。

$$(1 - \rho_1 L)(1 - \rho_{4,1} L^4) \Delta \Delta_4 \ln m1_t = \epsilon_t$$

フィットを実行します。

```
arima DS4.lnm1, ar(1) mar(1,4) noconstant nolog
```

ARIMA regression

Sample: 1961q2 - 2008q2

Number of obs = 189

Wald chi2(2) = 119.78

Log likelihood = 588.6689

Prob > chi2 = 0.0000

DS4.lnm1		Coef.	OPG Std. Err.	z	P> z	[95% Conf. Interval]	
ARMA							
	ar						
	L1.	.489277	.0538033	9.09	0.000	.3838245	.5947296
ARMA4							
	ar						
	L1.	-.4688653	.0601248	-7.80	0.000	-.5867076	-.3510229
/sigma		.0107075	.0004747	22.56	0.000	.0097771	.0116379

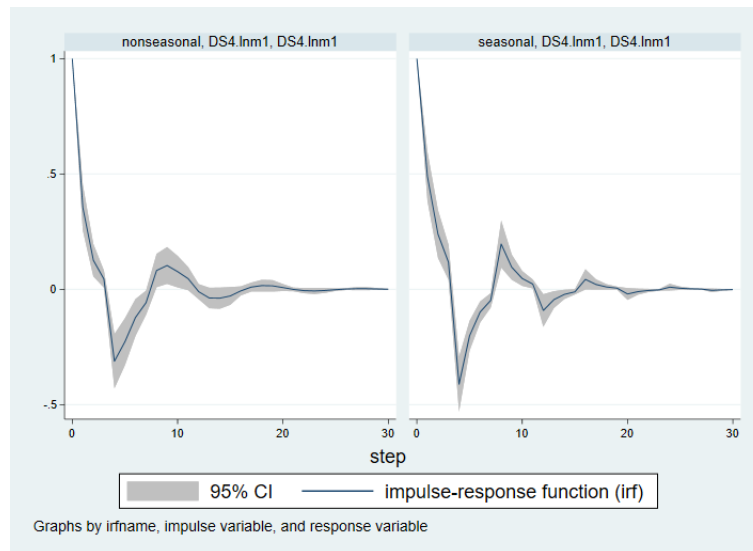
Note: The test of the variance against zero is one sided, and the two-sided confidence interval is truncated at zero.

IRF 結果を先程の `myirf.irf` ファイルに保存します。

```
irf create seasonal, step(30)
```

`myirf.irf` ファイルには2つの IRF 結果のセットが保存されています。`irf graph` コマンドを使用して、これらをグラフ化します。

```
irf graph irf
```

- IRF 関数の軌跡は似ています。各図は、**lnm1** へのショックが **lnm1** の一時的な振動を引き起こし、約 15 時点後に消滅することを示しています。

コレログラム

- **corrgram** コマンドは、自己相関、偏自己相関、およびかばん検定の統計量の表を作成します。また、自己相関と偏自己相関をプロットします。
- **ac** コマンドは、Bartlett の MA(q)過程の定義式に基づいた点ごとの信頼区間でコレログラム（自己相関のグラフ）を作成します。
- **pac** コマンドは、 $1/\sqrt{n}$ の標準誤差を使用して計算された信頼区間で偏コレログラム（偏自己相関のグラフ）を作成します。

例題 1

- 国際航空会社の乗客データセットを使用します (Box, Jenkins, and Reinsel 2008, Series G)。このデータセットには、1949 年から 1960 年までの国際航空会社の月間乗客数に関する 144 の観測値が含まれます。
- **corrgram** を使用して、自己相関と偏自己相関を一覧表示できます。

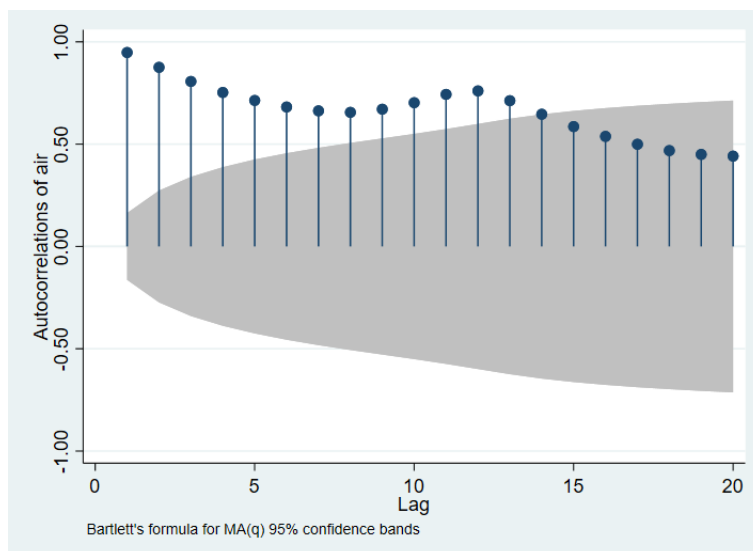
データセットをインポートして **corrgram** を実行します。

```
use https://www.stata-press.com/data/r16/air2  
  
corrgram air, lags(20)
```

LAG	AC	PAC	Q	Prob>Q	-1	0	1	-1	0	1
					[Autocorrelation]			[Partial Autocor]		
1	0.9480	0.9589	132.14	0.0000						
2	0.8756	-0.3298	245.65	0.0000						
3	0.8067	0.2018	342.67	0.0000						
4	0.7526	0.1450	427.74	0.0000						
5	0.7138	0.2585	504.8	0.0000						
6	0.6817	-0.0269	575.6	0.0000						
7	0.6629	0.2043	643.04	0.0000						
8	0.6556	0.1561	709.48	0.0000						
9	0.6709	0.5686	779.59	0.0000						
10	0.7027	0.2926	857.07	0.0000						
11	0.7432	0.8402	944.39	0.0000						
12	0.7604	0.6127	1036.5	0.0000						
13	0.7127	-0.6660	1118	0.0000						
14	0.6463	-0.3846	1185.6	0.0000						
15	0.5859	0.0787	1241.5	0.0000						
16	0.5380	-0.0266	1289	0.0000						
17	0.4997	-0.0581	1330.4	0.0000						
18	0.4687	-0.0435	1367	0.0000						
19	0.4499	0.2773	1401.1	0.0000						
20	0.4416	-0.0405	1434.1	0.0000						

ac コマンドで自己相関のグラフを作成します。

```
ac air, lags(20)
```



- データには、季節要素だけでなくトレンド要素も含まれている可能性があります。第一階差はトレンドの影響を軽減し、季節階差は季節性の制御に役立ちます。

トレンドと季節性を制御した後の偏自己相関をグラフ化します。`srv` オプションを使用して、標準化された残差分散を含めます。

```
pac DS12.air, lags(20) srv
```

