

**UNIVERSITA' DEGLI STUDI DI PADOVA**

FACOLTA' DI SCIENZE STATISTICHE

CORSO DI LAUREA IN STATISTICA E TECNOLOGIE  
INFORMATICHE



TESI DI LAUREA

**LA PREVISIONE DI SERIE STORICHE  
ECONOMICHE UTILIZZANDO LA  
TRASFORMAZIONE LOGARITMICA**

RELATORE: PROF.SSA LUISA BISAGLIA

LAUREANDO: NICOLA SABBADIN  
MATRICOLA: 560980 – STI

ANNO ACCADEMICO 2010-2011



## Sommario

|   |    |
|---|----|
| Sommario .....  | 3  |
| 1. Introduzione .....   | 5  |
| 2. Brevi Accenni sulle Serie Storiche Economiche .....                      | 7  |
| 3. I modelli ARIMA .....  | 11 |
| 3.1 Processo rumore bianco (white noise).....                               | 11 |
| 3.2 Processo a media mobile - $Ma(q)$ .....                                 | 12 |
| 3.3 Processo autoregressivo - $Ar(p)$ .....                                 | 13 |
| 3.4 Processo autoregressivo a media mobile - $Arma(p,q)$ .....              | 14 |
| 3.5 Processo autoregressivo integrato a media mobile - $Arima(p,d,q)$ ..... | 15 |
| 4. Brevi cenni sulla Trasformazione Logaritmica .....                       | 17 |
| 5. Log Trasformate di Serie Storiche .....                                  | 19 |
| 5.1. Processi Stazionari .....  | 21 |
| 5.2. Processi non Stazionari .....  | 23 |
| 5.3. Primo e secondo esperimento Monte Carlo .....                          | 25 |
| 5.4. Terzo esperimento Monte Carlo .....                                    | 28 |
| 6. Risultati Simulazioni .....  | 31 |
| 6.1 Risultati primo esperimento .....                                       | 31 |
| 6.2 Risultati secondo esperimento .....                                     | 33 |
| 6.3 Risultati terzo esperimento .....                                       | 34 |
| 7. Un Esempio con serie Eteroschedastiche .....                             | 37 |
| 8. Appendice A: Codice R utilizzato .....                                   | 43 |
| 9. Appendice B: Risultati simulazioni .....                                 | 53 |
| 10. Bibliografia .....  | 63 |



## 1. Introduzione

Per i modelli previsivi e le analisi economiche molte variabili vengono studiate mediante passaggio ai logaritmi. Nell'analisi delle serie storiche questa trasformazione è molto importante e molto utilizzata perché ha la proprietà di stabilizzare la varianza delle serie considerate, cosa molto utile nel caso la serie presenti proprietà eteroschedastiche.

In questo lavoro ci si propone di studiare, mediante esperimenti Monte Carlo, sotto quali condizioni il passaggio ai logaritmi è conveniente e quando invece può risultare dannoso per la precisione delle previsioni, per far questo vengono confrontate le previsioni delle serie originali con quelle delle serie trasformate.

Questo lavoro si basa, principalmente, sull'articolo di Helmut Lutkepohl e Fang Xu *The Role of the Log Transformations in Forecasting Economic Variables* (2009), al quale ci si attiene per le simulazioni riguardanti le serie a varianza costante.

Vengono simulate anche delle serie eteroschedastiche per studiare la validità della trasformazione logaritmica in questo caso.

I risultati evidenziano che il miglioramento nelle previsioni si ha quando il passaggio ai logaritmi effettivamente stabilizza la varianza della serie originale, se ciò non avviene la trasformazione logaritmica può notevolmente diminuire la precisione delle previsioni.



## **2. Brevi Accenni sulle Serie Storiche Economiche (Di Fonzo, Lisi, 2005)**

La misurazione dei fenomeni che evolvono nel tempo viene compiuta con lo scopo di osservare, ed eventualmente prevedere, le dinamiche future del fenomeno oggetto di studio.

Uno degli obiettivi dell'analisi delle serie storiche è quello di individuare le componenti che determinano l'evoluzione dei fenomeni del tempo.

Una serie storica si può definire una successione di dati numerici nella quale ogni dato è associato ad un particolare istante o intervallo di tempo.

I valori delle realizzazioni della generica variabile  $Y$  che formano una serie storica rilevati ad intervalli equidistanti, sono solitamente indicati con lettere minuscole ( $y_1, y_2, \dots, y_i, \dots, y_t$ ): l'indice "i" serve, pertanto, ad indicare la posizione nella serie storica dell'osservazione "Y".

Se in ciascun punto o intervallo di tempo viene osservato un solo fenomeno, la serie è detta "univariata". Se le variabili osservate sono più d'una la serie è detta multipla o "multivariata". Le serie storiche considerate in questo lavoro sono tuttavia esclusivamente del primo tipo.

In una serie storica l'ordine dei dati è materiale e non accidentale come nel caso di un campione casuale di osservazioni e il riconoscimento di tale ordine è una delle caratteristiche più importanti della serie stessa. È lecito presumere che vi sia una sorta di dipendenza tra osservazioni successive e che essa sia legata alla posizione dell'osservazione nella sequenza.

Lo studio di tale dipendenza e la conseguente possibilità di sfruttarla a fini previsivi rappresentano il cuore dell'analisi delle serie storiche.

Il parametro temporale “t”, che definisce l'ordinamento dei dati, appartiene ad un insieme parametrico “T” che può essere continuo o discreto. La cadenza temporale con cui osservare un fenomeno dipende dalla natura del fenomeno stesso. Per variabili di tipo economico, ad esempio, gli intervalli temporali più utilizzati sono giorni, settimane, mesi, trimestri e anni.

Un'altra importante distinzione è tra serie storiche deterministiche e stocastiche. Una serie storica  $y_t$  è detta deterministica se esiste una funzione  $f_t = f(t, y_{t-1}, y_{t-2}, \dots)$  tale per cui  $E(y_t - f_t)^2 = 0$ , per ogni  $t$  appartenente a  $T$ . Una serie deterministica può essere quindi prevista esattamente sulla base della storia passata.

La maggior parte delle serie storiche è tuttavia di tipo stocastico, nel senso che il futuro viene determinato dal passato solo in maniera parziale rendendo quindi impossibile l'elaborazione di previsioni del tutto prive di errore.

L'approccio classico tende ad individuare in una serie storica due componenti, una sistematica (o deterministica), ed una casuale. Tale impianto metodologico è finalizzato a scomporre la serie in singole componenti individuabili in maniera deterministica e quindi, “con certezza”; ciò che del dato osservato non è spiegato dalla parte deterministica è invece ritenuto un “residuo” dovuto al caso.

Quindi, un modello stocastico abbastanza generale per descrivere il processo generatore dei dati di una serie storica  $y_t$  relativa ad una variabile  $Y$  è dato da:

$$Y_t = f_t + u_t .$$



Secondo l'approccio classico dell'analisi delle serie storiche si suppone che esista una "legge di evoluzione temporale del fenomeno" rappresentata da  $f_t$ . La componente casuale  $u_t$  viene invece assunta a rappresentare l'insieme delle circostanze, ciascuna di entità trascurabile, che non si vogliono o possono considerare esplicitamente in  $Y_t$ .

Una volta individuate le componenti tendenziale, ciclica e stagionale, l'approccio classico permette di determinare in maniera "meccanica" il dato della serie storica a meno di una componente casuale. In tale approccio il tempo viene trattato direttamente o indirettamente come una variabile indipendente: è solamente col passare del tempo, difatti, che il trend si evolve, che il ciclo assume un andamento oscillante, e che la componente stagionale, se presente, assume andamenti regolari.

Il tempo è quindi la variabile fondamentale per l'origine delle "forze" che determinano il dato, e per tale motivo, è la principale variabile che spiega il fenomeno.

Invece l'approccio moderno, o stocastico, viene proposto da Box e Jenkins nel 1976. L'obiettivo finale dell'analisi è quello di comprendere la struttura probabilistica sottostante ai dati e di tentare di riprodurre il processo stocastico che si suppone abbia generato la serie storica, ossia l'insieme delle "forze" che hanno concorso a produrre il dato secondo leggi probabilistiche.

Nell'approccio moderno si ipotizza che  $f_t$  manchi o sia stata già "eliminata". L'attenzione è spostata sulla componente stocastica  $u_t$  che si ipotizza essere un

processo a componenti controllate del tipo  $u_t = g(Y_{t-1}, Y_{t-2}, \dots, \varepsilon_{t-1}, \varepsilon_{t-2}) + \varepsilon_t$ , dove  $\varepsilon_t$  è generata da un processo di *white noise* (illustrato nel capitolo 3).

Il processo stocastico non potrà mai essere individuato esattamente poiché abbiamo a disposizione un insieme finito di osservazioni (la serie storica), ciò che si può fare è tentare di “mimarlo” attraverso un modello.

Tuttavia, per poter identificare con una buona approssimazione un modello che “mimi” il processo generatore dei dati, occorre che questo possieda alcune caratteristiche: stazionarietà, invertibilità, il processo deve essere gaussiano e deve essere inoltre possibile che, dalla serie storica (ossia dal campione), si possano ricavare le proprietà dell'intero processo (i momenti calcolati sul campione devono convergere a quelli dell'intera popolazione).

I modelli utilizzati nei capitoli successivi per le simulazioni e lo studio delle serie storiche faranno riferimento all'approccio moderno.

### 3. I modelli ARIMA

#### 3.1 Processo rumore bianco (white noise)

Secondo la definizione data in “Serie storiche economiche” (T. Di Fonzo, F. Lisi, 2005) il processo *white noise* è uno dei più importanti processi stocastici. Si tratta di un processo  $\{\varepsilon_t\}$  che consiste di una sequenza di variabili casuali incorrelate a media e varianza costanti.

La funzione autocorrelazione di un processo *white noise* è:

3.1.1)

$$\rho_k = \begin{cases} 1 & k = 0 \\ 0 & k = \pm 1, \pm 2, \dots \end{cases} .$$

La funzione di autocorrelazione parziale,  $P_k$ , è identicamente uguale a  $\rho$  dal momento che le componenti sono serialmente incorrelate. Dalla 3.1.1) è immediato dedurre che tale processo è stazionario. Inoltre se le v.c.  $\varepsilon$  hanno distribuzione normale si parla di *white noise* gaussiano.

Per l'autocorrelazione empirica del processo vale il risultato  $\sqrt{n}\rho \sim N(0,1)$  e che quindi, a fini pratici, si possa considerare  $\rho \sim N(0, \frac{1}{n})$ . Questo risultato può essere utilizzato per stimare se i residui di un modello sono incorrelati. Il rifiuto di tale ipotesi ad un opportuno livello di significatività, comporta solitamente l'abbandono dell'intero modello, in quanto i residui non possono ritenersi solamente imputabili al caso.

### 3.2 Processo a media mobile - Ma(q)

Sia  $\{\varepsilon_t\}$  un processo *white noise* di media nulla e varianza  $\sigma_\varepsilon^2$ . Un processo a media mobile di ordine “q” è definito come:

3.2.1)

$$Y_t = \varepsilon_t - \theta_1\varepsilon_{t-1} - \dots - \theta_q\varepsilon_{t-q}$$

Con  $\theta_j$  (j = 1, ..., q) parametri costanti. Tale processo viene indicato con l’acronimo MA(q). Nella 3.2.1) la variabile  $Y_t$  è pensata come il risultato di una somma di impulsi casuali presenti e passati.

Utilizzando l’operatore ritardo B, tale che  $B^h Y_t = Y_{t-h}$ , h = 0, 1, 2, ..., e indicando con:

3.2.2)

$$\theta(B) = (1 - \theta_1 B - \dots - \theta_q B^q)$$

l’operatore polinomiale a media mobile, la 3.2.1) può essere scritta anche come:

3.2.3)

$$Y_t = \theta(B)\varepsilon_t.$$

La media di un processo MA(q) è pari a:

3.2.4)

$$E[Y_t] = E[\varepsilon_t - \theta_1\varepsilon_{t-1} - \dots - \theta_q\varepsilon_{t-q}] = 0$$

Poiché media, varianza e covarianza sono costanti e non dipendono dal tempo un processo MA(q) è sempre stazionario. Si dimostra inoltre (Hamilton, 1994) che il processo è invertibile se tutte le radici dell'equazione caratteristica  $\Theta(B) = 0$  sono, in modulo, maggiori di uno.

### 3.3 Processo autoregressivo - Ar(p)

Sia  $\{\varepsilon_t\}$  un processo *white noise* di media nulla e varianza  $\sigma_\varepsilon^2$ . Si dice che  $\{Y_t\}$  è un processo auto regressivo di ordine “p” e viene indicato con AR(p) se:

3.3.1)

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

con  $\phi_i$  ( $i = 0, \dots, p$ ) parametri costanti.

Nella 3.3.1) la variabile  $Y_t$  è pensata come il risultato di una somma pesata di valori passati e di uno “shock” casuale contemporaneo.

Indicando con:

3.3.2)

$$\phi(B) = (1 - \phi_1 B - \dots - \phi_p B^p)$$

L'operatore polinomiale autoregressivo, un processo AR(p) può essere scritto come:

3.3.3)

$$\phi(B)Y_t = \phi_0 + \varepsilon_t.$$

La media di un processo AR(p) è pari a:

3.3.4)

$$\mu = E[Y_t] = \frac{\phi_0}{1 - \phi_1 - \dots - \phi_p}$$

Questo implica che la media di  $Y_t$  è zero se e solo se  $\phi_0 = 0$ .

Il processo AR(p) è sempre invertibile, nella 3.3.1) infatti la variabile  $Y_t$  è già espressa in funzione dei suoi valori passati. Inoltre è stazionario se tutte le “p” radici dell’equazione caratteristica  $\phi(B) = 0$  sono, in modulo, maggiori di uno.

### 3.4 Processo autoregressivo a media mobile - Arma(p,q)

Nelle formule pure dei processi AR e MA si possono avere un gran numero di parametri. Una rappresentazione più parsimoniosa può essere ottenuta considerando un modello misto.

Si parte da un processo *white noise*  $\{\varepsilon_t\}$  di media nulla e varianza  $\sigma_\varepsilon^2$ . Si dice che  $\{Y_t\}$  è un processo autoregressivo a media mobile di ordine (p,q) e lo si indica con ARMA(p,q) se:

3.4.1)

$$Y_t - \sum_{i=1}^p \phi_i Y_{t-i} = \phi_0 + \varepsilon_t - \sum_{j=1}^q \theta_j \varepsilon_{t-j}$$

Ovviamente quando  $p = 0$  si torna ad avere un  $MA(q)$  e quando  $q = 0$  si ha un  $AR(p)$ . Modelli come questi sono stati proposti da Box e Jenkins (1976) come una classe standard per il trattamento di serie storiche.

Anche in questo caso la 3.4.1) può essere espressa in forma più compatta utilizzando gli operatori polinomiali:

3.4.2)

$$\phi(B)Y_t = \phi_0 + \theta(B)\varepsilon_t.$$

Un processo ARMA è stazionario se tutte le radici dell'equazione caratteristica  $\phi(B) = 0$  sono, in modulo, maggiori di uno, mentre è invertibile se le "q" radici dell'equazione  $\theta(B) = 0$  sono tutte, in modulo, maggiori di uno.

### **3.5 Processo autoregressivo integrato a media mobile - Arima(p,d,q)**

Box e Jenkins (1976) hanno proposto un'estensione dei processi ARMA che riguarda certi speciali tipi di processi non stazionari. Essi, cioè considerano processi non stazionari omogenei di grado "d" che possono essere resi stazionari mediante opportune trasformazioni.

Sia  $\{\varepsilon\}$  un processo *white noise* di media zero e varianza  $\sigma_\varepsilon^2$ . Si indichi con  $X_t$  la d-esima differenza di  $Y_t$ ,  $X_t = (1 - B)^d Y_t$ . Si dice che  $\{Y_t\}$  è un processo autoregressivo integrato a media mobile di ordine (p,d,q), e lo si indica con  $ARIMA(p,d,q)$  se  $\{X_t\}$  è un processo  $ARMA(p,q)$ .

Valgono le seguenti relazioni:

3.5.1)

$$\begin{aligned} X_t &= (1 - B)^d Y_t, \\ X_t &= \phi_0 + \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t - \sum_{j=1}^q \theta_j \varepsilon_{t-j}. \end{aligned}$$

Utilizzando l'operatore differenza ed i polinomi, rispettivamente, autoregressivo ed a media mobile, la 3.5.1) può essere riscritta in termini di  $Y_t$  :

3.5.2)

$$\phi(B)(1 - B)^d Y_t = \phi_0 + \theta(B)\varepsilon_t.$$

È evidente che per  $d = p = 0$  la classe dei processi ARIMA(p,d,q) si riduce a quella dei processi AR(p), per  $p = d = 0$  si ottiene la classe dei processi MA(q) e per  $d = 0$  si ha  $Y_t \sim$  ARMA(p,q).

Quando poi si considera il caso particolare  $p = d = q = 0$  si ottiene il processo *white noise*. È immediato anche riconoscere che un modello *random walk* non è altro che un ARIMA(0,1,0).



## 4. Brevi cenni sulla Trasformazione Logaritmica

In statistica talvolta si può essere interessati al trattamento di una trasformazione dei dati piuttosto che ai valori originari. In generale, un filtro è una funzione che trasforma una serie di dati in un'altra, con delle caratteristiche specifiche ricercate che ne rendono più semplice lo studio. Distinguiamo i filtri lineari dai filtri non lineari.

I primi sono combinazioni lineari dei valori di una serie, specularmente, i secondi sono combinazioni non lineari dei valori originari di una serie. Un esempio di filtro lineare è la media mobile semplice.

La trasformazione in scala logaritmica è invece un tipico esempio di filtro non lineare, utilizzato spesso nella rappresentazione grafica di particolari variabili. Talvolta accade difatti che i dati relativi ad un fenomeno presentino un valore molto piccolo della varianza all'inizio della serie, valore che va poi ad aumentare col passare del tempo. Vediamo un esempio:

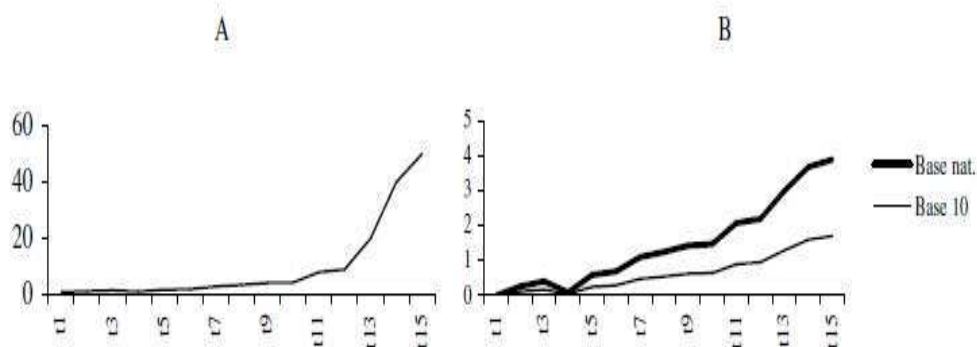


Figura 4.1: Esempio di serie storica in scala naturale (A) e in scala logaritmica (B).

La rappresentazione grafica attraverso la scala naturale non è in grado di porre il giusto rilievo sulla variazione dei valori più piccoli (*Figura 4.1, Grafico A*): per risolvere tale problema si suole utilizzare la trasformazione logaritmica dei dati della serie storica (*Figura 4.1, Grafico B*), il cui risultato grafico è l'amplificazione dei valori più piccoli, ed il graduale "schiacciamento" di quelli più grandi. E' ovvio che tale modalità operativa può essere utilizzata solamente nel caso di valori strettamente positivi.

Il Grafico B riporta un esempio di trasformazione logaritmica in base naturale e in base dieci. Come si può notare, la trasformazione più fedele alle caratteristiche del fenomeno originario è quella in base naturale.

## 5. Log Trasformate di Serie Storiche

Come evidenziato nell'articolo *The Role of the Log Transformations in Forecasting Economic Variables* (Helmut Lutkepohl, Fang Xu, 2009), soprattutto negli studi di serie storiche relative a variabili economiche c'è la diffusa opinione che, per un motivo o un altro, le trasformazioni logaritmiche siano importanti per uno studio corretto del modello.

Lavorando con serie temporali univariate la trasformazione logaritmica viene usata perché la trasformata può essere di miglior adattabilità ad un modello ARIMA standard, questo soprattutto perché le trasformazioni logaritmiche tendono a stabilizzare la varianza della serie.

Ovviamente, una volta ottenute le previsioni è possibile, in maniera molto semplice, tornare sulla scala della serie originale applicando la funzione esponenziale alle previsioni ottenute.

Naturalmente non mancano gli inconvenienti, ci sono vari aspetti che bisogna considerare per non incorrere in errori di previsione grossolani, dove i risultati avuti con la serie trasformata si possono discostare in modo radicale da quelli che dovrebbero essere ottenuti senza applicare alcuna trasformazione ai dati.

Infatti è risaputo che le trasformazioni non-lineari applicate alle previsioni ottimali di una qualunque variabile d'interesse potrebbero non essere uguali alle previsioni ottimali della variabile trasformata. A questi risultati sono arrivati Granger e Newbold nel loro articolo del 1976. In particolare, se si dispone delle previsioni dei logaritmi, convertirle in previsioni per la serie originale applicando la funzione esponenziale non sempre garantisce risultati ottimali.

Per questo motivo si andranno a confrontare differenti previsioni per variabili che tipicamente si considerano negli studi di Economia:

1. Previsioni mediante costruzione di modelli ARIMA per la variabile originale senza alcuna trasformazione: questo tipo di previsioni non è del tutto infondato poiché il modello ARIMA cattura la media condizionata e questo è importante per le stime puntuali. Le trasformazioni logaritmiche sono utili per stabilizzare la varianza e per questo sono determinanti in un secondo momento.
2. Previsioni sempre mediante modelli ARIMA basati sui logaritmi della serie, dove le previsioni per la serie originale si ottengono applicando la funzione esponenziale alle previsioni della serie logaritmica.
3. Le previsioni della serie logaritmica di cui al punto 2. vengono convertite in previsioni per la serie originale tramite una più complessa trasformazione che dovrebbe fornire risultati migliori in condizioni ideali.

Con questo esperimento si vuole confrontare la “performance” dei tre diversi tipi di previsione utilizzando delle tecniche di simulazione di tipo Monte Carlo.

## 5.1. Processi Stazionari

Sia  $x_t = \log y_t$  il logaritmo naturale della serie storica  $y_t$  e si supponga di generare  $x_t$  da un processo stazionario ARMA(p,q):

5.1.b)

$$\alpha(L)x_t = \nu + \theta(L)\varepsilon_t$$

dove “ $\nu$ ” è una costante,  $\alpha(L)$  e  $\theta(L)$  sono polinomi di ordine  $p$  e  $q$  scritti utilizzando l’operatore di ritardo e  $\varepsilon_t \sim N(0, \sigma^2)$  è il *white noise*.

Secondo Granger e Newbold (1976) la serie  $y_t$  è stazionaria se anche  $x_t$  mostra la stessa proprietà. Inoltre, sempre nello stesso articolo viene fatto notare che se  $x_t$  è generata da un processo MA(q) allora anche  $y_t$  è approssimabile ad un processo MA di ordine massimo  $q$ .

D’altro canto, invece, se  $x_t$  è generata da un processo misto ARMA(p,q) dove la componente auto regressiva non è trascurabile ( $p > 0$ ) le cose si fanno più complicate.

In ogni caso è possibile stimare un modello previsivo “ $h$  passi in avanti” per  $y_{(t+h)}$  dove  $y_t, y_{(t-1)}, y_{(t-2)}, \dots$ , sono i dati sui quali si costruisce il modello e  $y_{t+1}, y_{t+2}, \dots, y_{t+h}$  sono le previsioni condizionate all’informazione passata. Tutto questo partendo da un modello ARMA(p,q) opportunamente adattato alla variabile di studio  $y_t$ .

Le previsioni ottenute con questo metodo, cioè applicando il modello direttamente alla serie  $y_t$  sono denominate:

$$y_{t+h|t}^{lin}$$

ossia sono le previsioni lineari ottimali per  $y_{t+h}$  data la serie  $y_t$  alla quale applicare il miglior modello previsivo ARMA(p,q).

Un altro modo per ottenere delle previsioni  $y_{t+h}$  può essere partire dalla serie  $x_t$ . Dal momento che  $x_t$  è generata da un processo ARMA stazionario dove  $x_t, x_{t-1}, x_{t-2}, \dots$ , sono le osservazioni, a partire da queste è possibile ottenere le previsioni ottimali  $x_{t+1}, x_{t+2}, \dots, x_{t+h}$  fino a “h” passi in avanti.

Gli errori di previsione al passo “h” avranno ovviamente media zero e varianza che sarà denotata come  $\sigma_x^2(h)$ .

Le previsioni  $y_{t+h}$  per la serie  $y_t$  si possono ottenere ribaltando la trasformazione logaritmica:

5.1.c)

$$y_{t+h|t}^{nai} = \exp(x_{t+h|t})$$

Granger e Newbold (1976) dichiarano che queste previsioni sono “Naive”, “ingenue”, poiché non sono previsioni ottimali, dimostrando che quelle ottimali si ottengono “correggendo”  $x_{t+h}$  nella seguente maniera:

5.1.d)

$$y_{t+h|t}^{opt} = \exp\left(x_{t+h|t} + \frac{1}{2}\sigma_x^2(h)\right)$$

In seguito verranno comparate

$$y_{t+h|t}^{lin} \quad y_{t+h|t}^{nai} \quad y_{t+h|t}^{opt}$$

## 5.2. Processi non Stazionari

I modelli di serie storiche fin qui considerati sono tutti stazionari. Tuttavia, soprattutto per variabili economiche, è piuttosto frequente trovare serie per le quali è ragionevole ipotizzare un processo generatore non stazionario. Alcuni tipi di non-stazionarietà possono essere eliminati mediante differenziazione. In particolare si dice che un processo è non-stazionario omogeneo di grado “d” se diventa stazionario a seguito di “d” differenziazioni successive.

Come nel caso dei modelli ARMA possiamo usare gli stessi predittori e applicare gli stessi principi ai modelli ARIMA.

Nel prosieguo si tratterà quindi con modelli ARIMA considerando il caso di una sola integrazione: infatti sono i migliori esempi anche da un punto di vista pratico.

Una precisazione è tuttavia doverosa: se la variabile trasformata  $x_t = \log y_t$  è integrata la situazione non è di facile determinazione.

Di solito se  $y_t$  ha bisogno di essere trasformata per ottenere un processo ARIMA non è detto che abbia le stesse caratteristiche di una variabile I(1), come dichiarato nell’articolo di Granger e Hallman (1991) che ne studia le autocorrelazioni.

Perciò potrebbe accadere che  $y_t$  non sia classificata come una I(1) anche se

$$x_t = \log(y_t)$$

magari lo è.

Malgrado questo problema possa presentarsi negli esempi pratici e possa portare a scelte diverse nei procedimenti da adottare per ottenere le previsioni, in generale l'approccio primario sembra essere quello che dice che, partendo da un modello ARIMA, se  $x_t$  è classificata come I(1) allora c'è una buona probabilità che anche  $y_t = \exp x_t$  lo sia.

Quando si lavora con serie storiche economiche reali è bene assicurarsi che sia le serie originali che i logaritmi di queste abbiano lo stesso grado "d" di integrazione. Nelle simulazioni invece si andranno a studiare i livelli di "performance" dei predittori in un ambiente "controllato", quindi eventuali anomalie sono facilmente eliminabili.

Tramite un esperimento monte Carlo vogliamo, ancora una volta, valutare la bontà dei tre differenti predittori:

$$y_{t+h|t}^{lin} \quad y_{t+h|t}^{nai} \quad y_{t+h|t}^{opt}$$

Saranno considerate due diverse situazioni, nella prima  $x_t$  verrà generato da un processo ARI di modo che la varianza delle logaritmiche di  $y_t = \exp x_t$  sia omogenea. Nella seconda si genererà direttamente  $y_t$  sempre da un processo ARI,  $y_t$  alle quali saranno ugualmente applicate le trasformazioni logaritmiche anche se la varianza della serie è già omogenea.



### 5.3. Primo e secondo esperimento Monte Carlo

Si parte da un processo AR(1) per la simulazione di una serie stazionaria  $x_t$  alla quale verranno applicate le differenze prime per ottenere la serie  $x_t$  approssimata da un modello ARI(1,1) non stazionario.

In altre parole, utilizzando l'operatore differenziazione:

$$\Delta = (1 - L)$$

il processo generatore dati diventa il seguente:

5.3.a)

$$\Delta x_t = \nu + \rho \Delta x_{t-1} + \epsilon_t \quad t = 1, 2, \dots$$

Con  $x(0) = 0$ ,  $\rho = -0.9, -0.5, 0, 0.5, 0.9$ ,  $\nu$  è un termine costante che induce uno scostamento nei livelli della serie integrata  $x_t$  e  $\epsilon_t$  è indipendente e identicamente distribuito come una normale di media zero e varianza  $\sigma^2_{\epsilon}$ .

Vengono considerati campioni di ampiezza  $T = 40$  e  $T = 80$ , e vengono prodotte previsioni 10 passi in avanti per controllare la bontà del modello.

Inoltre vengono scartati i primi 100 valori della serie per alleviare l'effetto dei valori iniziali.

La variabile  $y_t$  viene calcolata applicando la funzione esponenziale alla serie  $x_t$  generata col sopracitato metodo.

Per simulare un approccio che sia utile negli esempi pratici verranno stimati solamente processi AR(p) a partire dalla serie  $x_t$ . Gli ordini dei modelli AR

verranno scelti ogni volta mediante il Criterio di Akaike (AIC), con ordine massimo uguale a 4 per le serie con  $T = 40$  e 6 per le serie con  $T = 80$ .

Le varianze per gli errori di previsione utili per il calcolo delle previsioni ottimali sono stimati a questa maniera: siano  $\alpha_1, \alpha_2, \dots, \alpha_{p+1}$  i coefficienti di un modello AR( $p + 1$ ). L'operatore AR risulta essere:

5.3.b)

$$1 - \hat{\alpha}_1 L - \dots - \hat{\alpha}_{p+1} L^{p+1} = \hat{\rho}(L)(1 - L)$$

Dove  $\rho(L)$  è il polinomio AR( $p$ ) della serie nelle differenze.

Per di più si pone:

5.3.c)

$$\hat{\sigma}_\epsilon^2 = (T - 1 - p)^{-1} \sum_{t=2}^T \hat{\epsilon}_t^2$$

come stimatore della varianza residua.

Quindi assegnando a  $\phi_0$  il valore 1, calcoliamo  $\phi_1, \phi_2, \dots$ , ricorsivamente tramite la formula:

5.3.d)

$$\hat{\phi}_i = \sum_{j=1}^{\min(i, p-1)} \hat{\phi}_{i-j} \hat{\alpha}_j$$

Fatto questo si può ottenere la varianza degli errori di previsione “al passo  $i$ -esimo” come:

5.3.e)

$$\hat{\sigma}_x^2(h) = \hat{\sigma}_\epsilon^2 \sum_{i=0}^{h-1} \hat{\phi}_i^2$$

Si calcolano quindi gli MSE (Mean Squared Error) relativi alle previsioni ottenute. Gli esperimenti verranno replicati più volte, ovviamente l'esperimento che disegna la soluzione migliore sarà quello che presenterà un MSE più basso. L'MSE altro non è che la media aritmetica degli "n" moduli delle differenze tra valori reali e valori previsti per ciascuna delle "n" serie replicate.

Quindi nel primo esperimento saranno confrontati gli MSE delle previsioni "Naive" con quelli delle previsioni lineari, nel secondo gli MSE ottimali con quelli "Naive".

Il confronto verrà fatto sui rapporti fra i differenti MSE e la bontà di un metodo previsivo rispetto ad un altro sarà data dal "peso" minore dell' MSE migliore.

Nel primo esperimento, a livello di codice R, sono stati scritte delle funzioni che svolgono esattamente quanto precedentemente illustrato, producendo come output finale i rapporti tra i corrispettivi MSE per i due differenti modelli previsivi (nel caso in esame MSE "Naive" su MSE lineare) per tutti i valori fissati di  $\rho$  (-0.9, -0.5, 0, 0.5, 0.9) e  $v$  (0, 0.01, 0.02).

Il processo viene inoltre ripetuto per due diversi valori della varianza delle serie simulate,  $\sigma^2 = 0.001$  e  $\sigma^2 = 0.0001$ .

Nel secondo esperimento verranno invece messi a confronto gli MSE delle previsioni ottimali con quelli delle previsioni "Naive".

Il procedimento è molto simile a quello visto per il primo esperimento, per questo le funzioni da utilizzare rimangono sostanzialmente le stesse. L'unica differenza è che si necessita di una funzione di simulazione specifica per la correzione dei valori previsti prima di passarli alla funzione esponenziale e ottenere così delle previsioni ottimali.

I valori di  $\rho$  sono posti uguali a quelli del primo esperimento, mentre viene analizzato il solo caso di  $v$  uguale a zero, dal momento che, in simulazioni analoghe svolte in precedenza, al variare di  $v$  i valori ottenuti rimanevano sostanzialmente gli stessi. Anche questa volta vengono esaminati due diversi valori per  $\sigma^2$  (0.001 e 0.0001).

I risultati delle simulazioni sono presenti nel prossimo capitolo.

#### **5.4. Terzo esperimento Monte Carlo**

Per analizzare quanto possa effettivamente essere rischiosa la trasformazione logaritmica per la precisione nelle previsioni nel caso non contribuisca a stabilizzare la varianza si procede con un altro esperimento dove la variabile di interesse  $y_t$  viene generata da un processo ARI.

Una volta fatto questo si passa ai logaritmi ottenendo la serie  $x_t$  sulla quale verranno fatte le previsioni esattamente come nell'esperimento precedente.

Più precisamente il processo di generazione dati di  $\Delta y_t$  è un AR(1):

5.4.a)

$$\Delta y_t = \nu + \rho \Delta y_{t-1} + \epsilon_t \quad t = 1, 2, \dots$$

Con  $y_0 = y_{-1} = 0$ .

Le  $x_t$  poi si ottengono passando ai logaritmi, ossia  $x_t = \log y_t$ .

L'ordine dei processi AR(p) stimati sarà scelto mediante il Criterio di Akaike con ordine massimo uguale a 4 per le serie con  $T = 40$  e ordine massimo 6 per le serie con  $T = 80$ .

Anche in questo esperimento verranno confrontati gli MSE delle previsioni "Naive" con quelli delle previsioni lineari, questo per valutare la convenienza della trasformazione logaritmica nel caso di una varianza stabile.

Sempre a livello di codice R per quanto riguarda le previsioni "Naive" la serie viene generata, con la stessa procedura fin qui utilizzata, da un ARI(1,1), si passa ai logaritmi dove sulla serie  $x_t$  viene costruito il solito modello autoregressivo AR(p) che genera previsioni. Da qui si ritornano le previsioni per la serie originale  $y_t$  mediante la funzione esponenziale.

Invece le previsioni lineari per la serie  $y_t$  vengono ottenute sulla serie stessa senza nessuna trasformazione.

Nel primo caso c'è anche la necessità che la serie generata sia positiva e che i suoi valori siano tutti maggiori o uguali ad 1, altrimenti il passaggio ai logaritmi potrebbe portare diversi problemi.

A livello di codice R sono stati posti dei controlli a tale scopo in modo da poter garantire le trasformazioni logaritmiche della serie originale  $y_t$  in tutta sicurezza. Come nelle simulazioni precedenti, le prime 100 osservazioni saranno scartate.

I valori utilizzati per il parametro  $\rho$  non cambiano, al parametro  $v$  vengono assegnati i valori 0.02 e 0.05 per assicurare, con un drift positivo, valori positivi per  $y(t)$  e rendere quindi agevole il passaggio alla serie logaritmica.

Anche in questo caso l'esperimento sarà replicato per due diversi valori della varianza del modello, 0.001 e 0.0001.

I risultati delle simulazioni sono presenti nel prossimo capitolo.

## 6. Risultati Simulazioni

Si riportano di seguito i risultati dei tre diversi esperimenti solo per il caso  $T = 80$ , dove  $T$  è la lunghezza della serie simulata, dal momento che i valori ottenuti per  $T = 40$  sono molto simili. In tutti i casi il parametro “ $p$ ” del modello AR( $p$ ) è identificato mediante il criterio di Akaike con ordine massimo 6. Il numero delle replicazioni è di 10000. Vengono presentati, per comodità, solamente le previsioni per  $h = 1$ ,  $h = 5$  e  $h = 10$ .

### 6.1 Risultati primo esperimento

Rapporto MSE delle previsioni “Naive” su MSE delle previsioni lineari per le serie simulate con varianza pari a 0.001. Processo generatore dei dati:  $\Delta x_t = v + \rho \Delta x_{t-1} + \varepsilon_t$ ,  $y_t = \exp x_t$ .

|      |     | $\sigma^2 = 0,001$ |       |       |       |       |
|------|-----|--------------------|-------|-------|-------|-------|
|      |     | $\rho =$           |       |       |       |       |
| $v$  | $h$ | -0,9               | -0,5  | 0     | 0,5   | 0,9   |
| 0    | 1   | 1,014              | 1,004 | 1,003 | 0,987 | 1,275 |
|      | 5   | 0,998              | 1,019 | 1,015 | 0,993 | 1,259 |
|      | 10  | 0,999              | 1,011 | 1,028 | 0,996 | 1,148 |
| 0,01 | 1   | 0,995              | 1,017 | 1,059 | 1,277 | 0,963 |
|      | 5   | 1,009              | 1,063 | 1,103 | 1,204 | 0,903 |
|      | 10  | 1,021              | 1,058 | 1,095 | 1,138 | 0,858 |
| 0,02 | 1   | 1,001              | 1,031 | 1,174 | 1,266 | 1,075 |
|      | 5   | 1,043              | 1,099 | 1,171 | 1,162 | 1,079 |
|      | 10  | 1,051              | 1,081 | 1,131 | 1,084 | 1,073 |

Tabella 6.1.1: previsioni a 1, 5 e 10 passi in avanti.

Rapporto MSE delle previsioni "Naive" su MSE delle previsioni lineari per le serie simulate con varianza pari a 0.0001. Processo generatore dei dati:  $\Delta x_t = v + \rho \Delta x_{t-1} + \varepsilon_t$ ,  $y_t = \exp x_t$ .

|      |    | $\sigma^2 = 0,0001$ |       |       |       |       |
|------|----|---------------------|-------|-------|-------|-------|
|      |    | $\rho =$            |       |       |       |       |
| v    | h  | -0,9                | -0,5  | 0     | 0,5   | 0,9   |
| 0    | 1  | 1,007               | 0,977 | 1,009 | 0,989 | 1,058 |
|      | 5  | 1,008               | 0,985 | 1,001 | 0,984 | 1,039 |
|      | 10 | 0,994               | 1,008 | 1,001 | 0,974 | 1,028 |
| 0,01 | 1  | 1,007               | 1,034 | 1,133 | 1,276 | 1,086 |
|      | 5  | 1,02                | 1,061 | 1,111 | 1,201 | 1,021 |
|      | 10 | 1,025               | 1,045 | 1,08  | 1,136 | 1,023 |
| 0,02 | 1  | 1,039               | 1,101 | 1,215 | 1,218 | 0,983 |
|      | 5  | 1,066               | 1,099 | 1,172 | 1,164 | 0,987 |
|      | 10 | 1,052               | 1,078 | 1,132 | 1,087 | 0,988 |

Tabella 6.1.2: previsioni a 1, 5 e 10 passi in avanti.



## 6.2 Risultati secondo esperimento

Rapporto MSE delle previsioni ottimali su MSE delle previsioni "Naive" per le serie simulate con varianza pari a 0.001. Processo generatore dei dati:  $\Delta x_t = v + \rho \Delta x_{t-1} + \varepsilon_t$ ,  $y_t = \exp x_t$ .

|   |    | $\sigma^2 = 0,001$ |       |       |       |       |
|---|----|--------------------|-------|-------|-------|-------|
|   |    | $\rho =$           |       |       |       |       |
| v | h  | -0,9               | -0,5  | 0     | 0,5   | 0,9   |
| 0 | 1  | 0,997              | 1,009 | 1,011 | 0,989 | 1,118 |
|   | 5  | 0,993              | 1,005 | 1,024 | 0,992 | 1,179 |
|   | 10 | 1,011              | 1,008 | 0,994 | 0,982 | 1,157 |

Tabella 6.2.1: previsioni a 1, 5 e 10 passi in avanti.

Rapporto MSE delle previsioni ottimali su MSE delle previsioni "Naive" per le serie simulate con varianza pari a 0.0001. Processo generatore dei dati:  $\Delta x_t = v + \rho \Delta x_{t-1} + \varepsilon_t$ ,  $y_t = \exp x_t$ .

|   |    | $\sigma^2 = 0,0001$ |       |       |       |       |
|---|----|---------------------|-------|-------|-------|-------|
|   |    | $\rho =$            |       |       |       |       |
| v | h  | -0,9                | -0,5  | 0     | 0,5   | 0,9   |
| 0 | 1  | 0,994               | 1,002 | 1,005 | 0,994 | 0,969 |
|   | 5  | 0,984               | 0,983 | 0,978 | 1,009 | 0,971 |
|   | 10 | 0,998               | 0,987 | 0,988 | 1,008 | 0,973 |

Tabella 6.2.2: previsioni a 1, 5 e 10 passi in avanti.

### 6.3 Risultati terzo esperimento

Rapporto MSE delle previsioni "Naive" su MSE delle previsioni lineari per le serie simulate con varianza pari a 0.001. Processo generatore dei dati:  $\Delta y_t = v + \rho \Delta y_{t-1} + \varepsilon_t$ ,  $x_t = \log y_t$ .

|      |    | $\sigma^2 = 0,001$ |       |       |       |       |
|------|----|--------------------|-------|-------|-------|-------|
|      |    | $\rho =$           |       |       |       |       |
| v    | h  | -0,9               | -0,5  | 0     | 0,5   | 0,9   |
| 0,02 | 1  | 1,019              | 1,051 | 1,217 | 1,637 | 4,211 |
|      | 5  | 1,061              | 1,104 | 1,204 | 1,516 | 3,016 |
|      | 10 | 1,052              | 1,088 | 1,166 | 1,411 | 2,282 |
| 0,05 | 1  | 1,095              | 1,241 | 1,463 | 2,096 | 4,685 |
|      | 5  | 1,135              | 1,221 | 1,386 | 1,858 | 1,973 |
|      | 10 | 1,114              | 1,179 | 1,311 | 1,651 | 2,261 |

Tabella 6.3.1: previsioni a 1, 5 e 10 passi in avanti.

MSE delle previsioni "Naive" su MSE delle previsioni lineari per le serie simulate con varianza pari a 0.0001. Processo generatore dei dati:  $\Delta y_t = v + \rho \Delta y_{t-1} + \varepsilon_t$ ,  $x_t = \log y_t$ .

|      |    | $\sigma^2 = 0,0001$ |       |       |       |       |
|------|----|---------------------|-------|-------|-------|-------|
|      |    | $\rho =$            |       |       |       |       |
| v    | h  | -0,9                | -0,5  | 0     | 0,5   | 0,9   |
| 0,02 | 1  | 1,049               | 1,114 | 1,235 | 1,631 | 4,211 |
|      | 5  | 1,066               | 1,102 | 1,197 | 1,519 | 3,016 |
|      | 10 | 1,055               | 1,084 | 1,165 | 1,412 | 2,282 |
| 0,05 | 1  | 1,161               | 1,251 | 1,459 | 2,097 | 4,686 |
|      | 5  | 1,151               | 1,125 | 1,384 | 1,857 | 2,973 |
|      | 10 | 1,124               | 1,177 | 1,311 | 1,651 | 2,261 |

Tabella 6.3.2: previsioni a 1, 5 e 10 passi in avanti.

Come si può vedere dai risultati delle simulazioni Monte Carlo la tendenza generale sembra indicare che la trasformazione logaritmica, nella situazione di omoschedasticità per la serie considerata, peggiora le previsioni invece di migliorarle.

Il primo esperimento mostra come, a parte in qualche caso in cui  $v$  è nullo e i valori sono vicini a uno, le previsioni lineari siano mediamente migliori di quelle effettuate sui logaritmi e all'aumentare di  $v$  aumenti anche il divario tra i due modelli proposti, sempre a favore di quello lineare.

Nel secondo esperimento, dove vengono messi a confronto previsioni ottimali con previsioni "Naive" i valori sono tutti molto vicini a uno, in alcuni casi sembra che le ottimali siano migliori, ma il miglioramento sembra non essere mai più di tanto significativo, indice del fatto che per le previsioni di serie omoschedastiche i due metodi bene o male si equivalgono.

Il terzo esperimento emette invece un verdetto univoco: la trasformazione logaritmica peggiora le previsioni in confronto al metodo previsivo lineare. Anzi, all'aumentare sia del parametro  $v$  sia di  $p$  aumenta anche il divario tra i due diversi metodi previsivi, sempre a favore di quello lineare. Risultato: la trasformazione logaritmica è addirittura dannosa.

Non sembrano esserci differenze significative tra le serie simulate con varianza uguale a 0.001 e quelle simulate con varianza uguale a 0.0001. Altro risultato interessante, sempre nel terzo esperimento, è che il divario tra previsione lineare e previsione "Naive" con trasformazione logaritmica sembra diminuire all'aumentare dell'orizzonte previsivo.



## 7. Un Esempio con serie Eteroschedastiche

Helmut Lutkepohl e Fang Xu, nel loro articolo *The Role of the Log Transformations in Forecasting Economic Variables* (2009), prendono in esame un insieme di serie storiche economiche e calcolano i tre diversi tipi di previsione visti nei capitoli precedenti per ciascuna di queste. Lo studio da loro fatto ha come scopo capire sotto quali condizioni l'utilizzo della trasformazione logaritmica migliora le previsioni per le serie eteroschedastiche.

Il risultato a cui sono giunti è che le previsioni basate sul passaggio ai logaritmi sono migliori di quelle lineari solamente nel caso la trasformazione logaritmica stabilizzi la varianza della serie.

Se ciò non accade il metodo di previsione lineare è migliore rispetto a quello che utilizza i logaritmi che, in questo caso, può risultare addirittura pericoloso per la precisione delle previsioni.

Quello che si vuole fare in questo capitolo è generare e studiare mediante simulazioni Monte Carlo delle serie eteroschedastiche simili a molte serie economiche osservabili nelle esperienze pratiche, con una varianza dei dati piccola all'inizio della serie che aumenta visibilmente da un certo punto in avanti.

Verranno considerate delle serie generate da un modello AR(1) per differenti valori del parametro  $\rho$  (-0.9, -0.5, 0, 0.5, 0.9). Sulle serie così ottenute verranno stimati dei modelli AR(p), come nei precedenti esperimenti, togliendo le ultime cinque osservazioni che in questo caso serviranno per confrontare i risultati.

Per ciascuna delle serie considerate verranno calcolati due modelli AR(p): uno sulla serie originale l'altro sulla trasformazione logaritmica di questa. L'ordine massimo è posto uguale a 4 e verrà identificato con il criterio di Akaike.

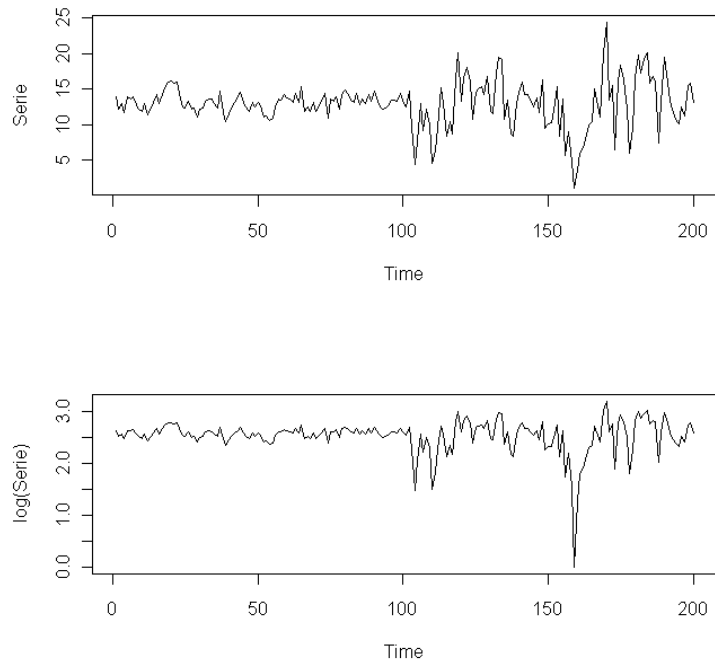
L'esperimento consiste, fissato il parametro  $\rho$ , nel replicare 10000 volte la serie e calcolare gli MSE per ciascun tipo di previsione (lineare o logaritmica) fino a 5 passi in avanti in maniera del tutto analoga a quanto fatto nei capitoli precedenti.

I risultati ottenuti vengono presentati come il rapporto tra MSE "Naive" e MSE lineare per ciascun valore del parametro  $\rho$ .

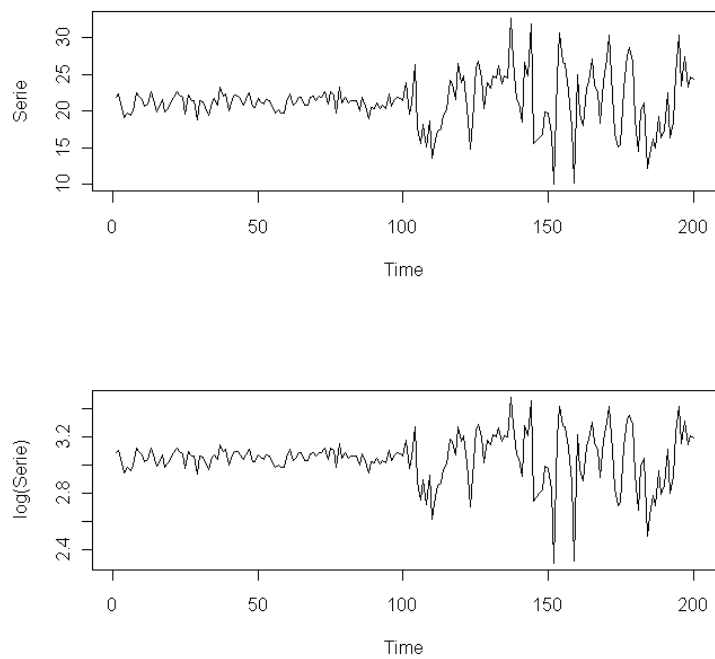
Viene inoltre considerato solo il caso delle previsioni "Naive" dal momento che Helmut Lutkepohl e Fang Xu nel loro articolo non riscontrano differenze significative nell'utilizzo delle previsioni ottimali.

L'esperimento viene ripetuto due volte, per serie che saranno indicate con "A" e serie che saranno indicate con "B". Nel secondo caso i valori assunti dalle osservazioni sono molto più grandi rispetto al nel primo, in modo da ottenere, con il passaggio ai logaritmi, dei valori abbastanza distanti da zero.

Nel caso "B" la trasformazione logaritmica sembra stabilizzare meglio la varianza come si può vedere dai grafici riportati in figura 7.1 e 7.2.



*Figura 7.1: Esempio di serie di tipo A e relativa serie logaritmica.*



*Figura 7.2: Esempio di serie di tipo B e relativa serie logaritmica.*

I risultati delle simulazioni sono riportati nelle seguenti tabelle:

Rapporto MSE delle previsioni "Naive" su MSE delle previsioni lineari per le serie simulate (tipo A)

Processo generatore dei dati:  $y_t = \rho y_{t-1} + \varepsilon_t$ ,  $x_t = \log y_t$ .

|        | h | 1      | 2      | 3      | 4      | 5      |
|--------|---|--------|--------|--------|--------|--------|
| $\rho$ |   |        |        |        |        |        |
| -0,9   |   | 1,3449 | 1,0549 | 1,0933 | 1,0344 | 1,0406 |
| -0,5   |   | 1,0694 | 1,0115 | 1,0179 | 1,0182 | 1,0148 |
| 0      |   | 1,0324 | 1,1015 | 1,0151 | 1,0245 | 1,0354 |
| 0,5    |   | 1,0286 | 1,0124 | 1,0186 | 1,0117 | 1,0111 |
| 0,9    |   | 1,1167 | 1,0916 | 1,0621 | 1,0457 | 1,0685 |

Tabella 7.1: previsioni da 1 a 5 passi in avanti.

Rapporto MSE delle previsioni "Naive" su MSE delle previsioni lineari per le serie simulate (tipo B)

Processo generatore dei dati:  $y_t = \rho y_{t-1} + \varepsilon_t$ ,  $x_t = \log y_t$ .

|        | h | 1      | 2      | 3      | 4      | 5      |
|--------|---|--------|--------|--------|--------|--------|
| $\rho$ |   |        |        |        |        |        |
| -0,9   |   | 0,9983 | 0,9998 | 0,9977 | 1,0001 | 0,9965 |
| -0,5   |   | 0,9987 | 0,9997 | 0,9999 | 0,9995 | 1,0002 |
| 0      |   | 0,9994 | 1,0001 | 0,9996 | 0,9986 | 0,9982 |
| 0,5    |   | 0,9998 | 0,9996 | 0,9989 | 0,9999 | 1,0001 |
| 0,9    |   | 0,9982 | 0,9998 | 0,9998 | 1,0004 | 0,9996 |

Tabella 7.2: previsioni da 1 a 5 passi in avanti.



Dai risultati si può vedere come per le serie di tipo A (Tabella 7.1) la trasformazione logaritmica abbia un MSE mediamente più grande di quella lineare. La situazione migliora nel caso delle serie di tipo B (Tabella 7.2) dove tra MSE “Naive” e MSE lineari non emergono differenze significative.

Nel secondo caso l’applicazione della trasformazione logaritmica sembra portare a previsioni migliori rispetto al primo caso dal momento che sembra esserci una migliore stabilizzazione della varianza.

Questo risultato è in linea con le conclusioni dell’articolo di Helmut Lutkepohl e Fang Xu. Quindi, per le previsioni di serie storiche dove la varianza non è omogenea, considerare il caso della trasformazione logaritmica può essere una valida alternativa. Nel caso il passaggio ai logaritmi non dovesse stabilizzare la varianza le previsioni lineari sono nettamente migliori.

Di volta in volta è quindi necessario controllare la bontà del modello previsivo scelto, non sempre la procedura teoricamente migliore è la migliore anche nella pratica. Utilizzare le ultime osservazioni di una serie storica per verificare la bontà del un modello previsivo scelto è sempre una buona prassi prima di passare al calcolo delle previsioni vere e proprie.



## 8. Appendice A: Codice R utilizzato

```
simulazione1<-function(ro,ni,T,sigma_quadro,p){
  if(ro==0){
    x.star<-rnorm(100+(T-1)+10, mean=ni, sd=sqrt(sigma_quadro))
  }
  else{
    x.star<-arima.sim(n=100+(T-1)+10, list(order=c(1,0,0),ar=ro), sd=sqrt(sigma_quadro),
    mean=(ni)/(1-ro))
  }
  x.star.1<-x.star[101:(100+(T-1)+10)]
  x<-diffinv(x.star.1)
  x1<-(x[1:T])
  x2<-(x[(T+1):(T+10)])
  y1<-exp(x1)
  y2<-exp(x2)
  m<-ar(y1,order.max=p)
  l3.predict.1<-predict(m,n.ahead=10)
  l3.predict<-l3.predict.1$pred
  y3<-
  c(l3.predict[1],l3.predict[2],l3.predict[3],l3.predict[4],l3.predict[5],l3.predict[6],l3.pr
  edict[7],l3.predict[8],l3.predict[9],l3.predict[10])
  diff<-y2-y3
  diff
}
#####
simulazione2<-function(ro,ni,T,sigma_quadro,p){
  if(ro==0){
    x.star<-rnorm(100+(T-1)+10, mean=ni, sd=sqrt(sigma_quadro))
  }
  else{
    x.star<-arima.sim(n=100+(T-1)+10, list(order=c(1,0,0),ar=ro), sd=sqrt(sigma_quadro),
    mean=(ni)/(1-ro))
  }
}
```

```

x.star.1<-x.star[101:(100+(T-1)+10)]
x<-diffinv(x.star.1)
x1<-(x[1:T])
x2<-(x[(T+1):(T+10)])
y2<-exp(x2)
m<-ar(x1,order.max=p)
l3.predict.1<-predict(m,n.ahead=10)
l3.predict<-l3.predict.1$pred

x3<-
c(l3.predict[1],l3.predict[2],l3.predict[3],l3.predict[4],l3.predict[5],l3.predict[6],l3.pr
edict[7],l3.predict[8],l3.predict[9],l3.predict[10])

y3<-exp(x3)
diff<-y2-y3
diff
}
#####
simulazione3<-function(ro,ni,T,sigma_quadro,p){
done<-FALSE
while(!done){
if(ro==0){
x.star<-rnorm(100+(T-1)+10, mean=ni, sd=sqrt(sigma_quadro))
}
else{
x.star<-arima.sim(n=100+(T-1)+10,list(order=c(1,0,0),ar=ro),sd=sqrt(sigma_quadro),
mean=(ni)/(1-ro))
}
x.star.1<-x.star[101:(100+(T-1)+10)]
x<-diffinv(x.star.1)
x1<-(x[1:T])
x2<-(x[(T+1):(T+10)])
y2<-exp(x2)
m<-ar(x1,order.max=p)
l3.predict.1<-predict(m,n.ahead=10)
l3.predict<-l3.predict.1$pred

```

```

x3<-
c(l3.predict[1],l3.predict[2],l3.predict[3],l3.predict[4],l3.predict[5],l3.predict[6],l3.pr
edict[7],l3.predict[8],l3.predict[9],l3.predict[10])

alpha<-m$alpha
P<-m$order-1
fi<-rep(1,10)
fi[1]=1
for(i in 2:10){
gei<-rep(0,10)
for(j in 1:(min(i,P+1))){
gei[j]=fi[i+1-j]*alpha[j]
}
fi[i]<-sum(gei)
}
res1<-m$resid
res<-na.omit(res1)
sigma2<-((sum(res^2))/(T-1-P))
sigma.h<-rep(0,10)
for(i in 1:10){
somma<-rep(0,10)
for(j in 1:i){
somma[j]<-(fi[j])^2
}
sigma.h[i]<-sigma2*sum(somma)
}
y3<-exp(x3+0.5*sigma.h)
diff1<-y2-y3
diff<-na.omit(diff1)
if(length(diff)==10) done<-TRUE
}
diff}

#####

simulazione4<-function(ro,ni,T,sigma_quadro,p){
done<- FALSE

```

```

while (!done) {
  if(ro==0){
    x.star<-rnorm(100+(T-1)+10, mean=ni, sd=sqrt(sigma_quadro))
  } else{
    x.star<-arima.sim(n=100+(T-1)+10, list(order=c(1,0,0),ar=ro), sd=sqrt(sigma_quadro),
    mean=(ni)/(1-ro))
  }
  x.star.1<-x.star[101:(100+(T-1)+10)]
  x.1<-diffinv(x.star.1)
  x<-x.1+1
  x.min<-min(x)
  if (x.min > 0)
    done<- TRUE
  }
  z1<-(x[1:T])
  z2<-(x[(T+1):(T+10)])
  l1<-log(z1)
  m<-ar(l1,order.max=p)
  l3.predict.1<-predict(m,n.ahead=10)
  l3.predict<-l3.predict.1$pred
  l3<-
  c(l3.predict[1],l3.predict[2],l3.predict[3],l3.predict[4],l3.predict[5],l3.predict[6],l3.pr
  edict[7],l3.predict[8],l3.predict[9],l3.predict[10])
  z3<-exp(l3)
  diff<-z2-z3
  diff
  }
#####
simulazione5<-function(ro,ni,T,sigma_quadro,p){
  if(ro==0){
    x.star<-rnorm(100+(T-1)+10, mean=ni, sd=sqrt(sigma_quadro))
  }
  else{
    x.star<-arima.sim(n=100+(T-1)+10, list(order=c(1,0,0),ar=ro), sd=sqrt(sigma_quadro),
    mean=(ni)/(1-ro))
  }
}

```

```

}
x.star.1<-x.star[101:(100+(T-1)+10)]
x<-diffinv(x.star.1)
z1<-(x[1:T])
z2<-(x[(T+1):(T+10)])
m<-ar(z1,order.max=p)
z3.predict.1<-predict(m,n.ahead=10)
z3.predict<-z3.predict.1$pred
z3<-
c(z3.predict[1],z3.predict[2],z3.predict[3],z3.predict[4],z3.predict[5],z3.predict[6],z3.
predict[7],z3.predict[8],z3.predict[9],z3.predict[10])
diff<-z2-z3
diff
}
#####
sim.montecarlo1<-function(ro,ni,T,sigma_quadro,p,n){
aa<-matrix(rep(0,n*10),n,10)
for(i in 1:n){
bb<-simulazione1(ro,ni,T,sigma_quadro,p)
aa[i,1]<-bb[1]
aa[i,2]<-bb[2]
aa[i,3]<-bb[3]
aa[i,4]<-bb[4]
aa[i,5]<-bb[5]
aa[i,6]<-bb[6]
aa[i,7]<-bb[7]
aa[i,8]<-bb[8]
aa[i,9]<-bb[9]
aa[i,10]<-bb[10]
}
MSE<-c( mean(abs(aa[,1])), mean(abs(aa[,2])), mean(abs(aa[,3])), mean(abs(aa[,4])),
mean(abs(aa[,5])), mean(abs(aa[,6])), mean(abs(aa[,7])), mean(abs(aa[,8])),
mean(abs(aa[,8])), mean(abs(aa[,10])))
MSE
}

```

```
#####
sim.montecarlo2<-function(ro,ni,T,sigma_quadro,p,n){
aa<-matrix(rep(0,n*10),n,10)
for(i in 1:n){
bb<-simulazione2(ro,ni,T,sigma_quadro,p)
aa[i,1]<-bb[1]
aa[i,2]<-bb[2]
aa[i,3]<-bb[3]
aa[i,4]<-bb[4]
aa[i,5]<-bb[5]
aa[i,6]<-bb[6]
aa[i,7]<-bb[7]
aa[i,8]<-bb[8]
aa[i,9]<-bb[9]
aa[i,10]<-bb[10]
}
MSE<-c( mean(abs(aa[,1])), mean(abs(aa[,2])), mean(abs(aa[,3])), mean(abs(aa[,4])),
mean(abs(aa[,5])), mean(abs(aa[,6])), mean(abs(aa[,7])), mean(abs(aa[,8])),
mean(abs(aa[,9])), mean(abs(aa[,10])))
MSE
}
#####
sim.montecarlo3<-function(ro,ni,T,sigma_quadro,p,n){
aa<-matrix(rep(0,n*10),n,10)
for(i in 1:n){
bb<-simulazione3(ro,ni,T,sigma_quadro,p)
aa[i,1]<-bb[1]
aa[i,2]<-bb[2]
aa[i,3]<-bb[3]
aa[i,4]<-bb[4]
aa[i,5]<-bb[5]
aa[i,6]<-bb[6]
aa[i,7]<-bb[7]
aa[i,8]<-bb[8]

```



```

aa[i,9]<-bb[9]

aa[i,10]<-bb[10]

}

MSE<-c( mean(abs(aa[,1])), mean(abs(aa[,2])), mean(abs(aa[,3])), mean(abs(aa[,4])),
mean(abs(aa[,5])), mean(abs(aa[,6])), mean(abs(aa[,7])), mean(abs(aa[,8])),
mean(abs(aa[,8])), mean(abs(aa[,10])))

MSE

}

#####

sim.montecarlo4<-function(ro,ni,T,sigma_quadro,p,n){
aa<-matrix(rep(0,n*10),n,10)

for(i in 1:n){

bb<-simulazione4(ro,ni,T,sigma_quadro,p)

aa[i,1]<-bb[1]

aa[i,2]<-bb[2]

aa[i,3]<-bb[3]

aa[i,4]<-bb[4]

aa[i,5]<-bb[5]

aa[i,6]<-bb[6]

aa[i,7]<-bb[7]

aa[i,8]<-bb[8]

aa[i,9]<-bb[9]

aa[i,10]<-bb[10]

}

MSE<-c( mean(abs(aa[,1])), mean(abs(aa[,2])), mean(abs(aa[,3])), mean(abs(aa[,4])),
mean(abs(aa[,5])), mean(abs(aa[,6])), mean(abs(aa[,7])), mean(abs(aa[,8])),
mean(abs(aa[,8])), mean(abs(aa[,10])))

MSE

}

#####

sim.montecarlo5<-function(ro,ni,T,sigma_quadro,p,n){
aa<-matrix(rep(0,n*10),n,10)

for(i in 1:n){

bb<-simulazione5(ro,ni,T,sigma_quadro,p)

aa[i,1]<-bb[1]

```

```

aa[i,2]<-bb[2]
aa[i,3]<-bb[3]
aa[i,4]<-bb[4]
aa[i,5]<-bb[5]
aa[i,6]<-bb[6]
aa[i,7]<-bb[7]
aa[i,8]<-bb[8]
aa[i,9]<-bb[9]
aa[i,10]<-bb[10]
}
MSE<-c( mean(abs(aa[,1])), mean(abs(aa[,2])), mean(abs(aa[,3])), mean(abs(aa[,4])),
mean(abs(aa[,5])), mean(abs(aa[,6])), mean(abs(aa[,7])), mean(abs(aa[,8])),
mean(abs(aa[,8])), mean(abs(aa[,10])))
MSE
}
#####
serie.sim1<-function(T,sigma_quadro,p,n){
ro<-c(-0.9,-0.5,0,0.5,0.9)
ni<-c(0,0.01,0.02)
aa<-matrix(rep(0,15*10),15,10)
z<-1
for(j in 1:3){
for(i in 1:5){
a<-sim.montecarlo1(ro[i],ni[j],T,sigma_quadro,p,n)
b<-sim.montecarlo2(ro[i],ni[j],T,sigma_quadro,p,n)
bb<-b/a
aa[z,1]<-bb[1]
aa[z,2]<-bb[2]
aa[z,3]<-bb[3]
aa[z,4]<-bb[4]
aa[z,5]<-bb[5]
aa[z,6]<-bb[6]
aa[z,7]<-bb[7]
aa[z,8]<-bb[8]

```

```

aa[z,9]<-bb[9]
aa[z,10]<-bb[10]
z<-z+1
}
}
aa
}
#####
serie.sim2<-function(T,sigma_quadro,p,n){
ro<-c(-0.9,-0.5,0,0.5,0.9)
aa<-matrix(rep(0,5*10),5,10)
z<-1
for(i in 1:5){
a<-sim.montecarlo2(ro[i],0,T,sigma_quadro,p,n)
b<-sim.montecarlo3(ro[i],0,T,sigma_quadro,p,n)
bb<-b/a
aa[z,1]<-bb[1]
aa[z,2]<-bb[2]
aa[z,3]<-bb[3]
aa[z,4]<-bb[4]
aa[z,5]<-bb[5]
aa[z,6]<-bb[6]
aa[z,7]<-bb[7]
aa[z,8]<-bb[8]
aa[z,9]<-bb[9]
aa[z,10]<-bb[10]
z<-z+1
}
}
aa
}
#####
serie.sim3<-function(T,sigma_quadro,p,n){

```

```

ro<-c(-0.9,-0.5,0,0.5,0.9)
ni<-c(0.02,0.05)
aa<-matrix(rep(0,10*10),10,10)
z<-1
for(j in 1:2){
for(i in 1:5){
a<-sim.montecarlo4(ro[i],ni[j],T,sigma_quadro,p,n)
b<-sim.montecarlo5(ro[i],ni[j],T,sigma_quadro,p,n)
bb<-a/b
aa[z,1]<-bb[1]
aa[z,2]<-bb[2]
aa[z,3]<-bb[3]
aa[z,4]<-bb[4]
aa[z,5]<-bb[5]
aa[z,6]<-bb[6]
aa[z,7]<-bb[7]
aa[z,8]<-bb[8]
aa[z,9]<-bb[9]
aa[z,10]<-bb[10]
z<-z+1
}
}
aa
}

```

## 9. Appendice B: Risultati simulazioni

### Esperimento 1: MSE "Naive" / MSE lineari

T=40, sigma\_quadro=0.001 p=4 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

0.9984543 1.0020931 0.9837043 1.000478 1.429889

0.9957893 1.0077107 0.9835872 1.005860 1.381532

0.9871277 0.9856036 0.9914436 1.008109 1.327158

1.0052529 1.0075480 0.9885617 1.003883 1.282600

0.9947317 1.0000028 0.9967025 1.007431 1.242698

1.0172664 1.0081822 0.9935518 1.010527 1.213091

0.9950034 1.0063628 0.9903423 1.018580 1.189191

1.0209524 1.0117088 0.9937360 1.026572 1.168071

1.0209524 1.0117088 0.9937360 1.026572 1.168071

1.0202204 1.0023361 0.9854472 1.033086 1.139014

nu=0.01, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.0057396 1.021192 1.061518 1.195851 1.0289346

0.9991699 1.027158 1.066325 1.180638 0.9759023

1.0142982 1.032988 1.065197 1.162435 0.9614251

0.9975846 1.034727 1.066685 1.145972 0.9569506

1.0320586 1.031475 1.068916 1.132309 0.9569123

1.0056839 1.028416 1.063219 1.118037 0.9593156

1.0360418 1.029358 1.058232 1.106083 0.9614912

1.0059518 1.022787 1.055455 1.096565 0.9641887

1.0059518 1.022787 1.055455 1.096565 0.9641887

1.0063760 1.017133 1.056267 1.079397 0.9766838

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.023368 1.050143 1.116880 1.277285 1.034650

1.043301 1.075328 1.112285 1.235027 1.037271

1.019659 1.066104 1.102008 1.201015 1.032759

1.048353 1.063075 1.089084 1.173131 1.029142

1.023803 1.062502 1.079935 1.150098 1.026235

1.041440 1.053826 1.072572 1.131578 1.025335

1.027759 1.049909 1.063118 1.117297 1.027013

1.041567 1.049987 1.057226 1.104428 1.032690

1.041567 1.049987 1.057226 1.104428 1.032690

1.033613 1.044194 1.047069 1.084410 1.064240

T=40 sigma\_quadro=0.0001 p=4 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.0164933 0.9823566 1.009705 1.013470 1.0079775

0.9897570 0.9963638 1.016243 1.012846 1.0050176

1.0135629 0.9873184 1.013344 1.012603 0.9998727

0.9963244 0.9862287 1.017728 1.010150 0.9967140

1.0105420 0.9956619 1.018167 1.014008 0.9932079

0.9942723 0.9889546 1.025155 1.015995 0.9927561

1.0065791 1.0017021 1.019661 1.012566 0.9925793

1.0071248 0.9968197 1.011111 1.011280 0.9919980

1.0071248 0.9968197 1.011111 1.011280 0.9919980

1.0031677 0.9908801 1.014885 1.007919 0.9884771

nu=0.01, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.017760 1.028005 1.069628 1.206039 1.077411

1.028230 1.032967 1.058566 1.185145 1.023423

1.017219 1.033952 1.056024 1.165156 1.011924

1.022523 1.025436 1.049770 1.147866 1.009635

1.018583 1.020192 1.044632 1.133266 1.008959

1.019169 1.020937 1.041771 1.119962 1.008700

1.012053 1.018066 1.038627 1.108248 1.008738

1.017148 1.017951 1.036192 1.098112 1.008833

1.017148 1.017951 1.036192 1.098112 1.008833

1.018848 1.013019 1.031720 1.081561 1.009352

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.040697 1.063563 1.121331 1.258913 1.006097

1.043643 1.050062 1.110252 1.219780 1.010070

1.039674 1.049408 1.100569 1.187224 1.009967

1.036894 1.047840 1.091088 1.160043 1.010103

1.032815 1.042019 1.083740 1.137204 1.010110

1.028753 1.037740 1.077175 1.118086 1.010406

1.030607 1.035080 1.071061 1.101758 1.010620

1.025070 1.031114 1.065973 1.088520 1.010854

1.025070 1.031114 1.065973 1.088520 1.010854

1.021273 1.026204 1.054366 1.067163 1.011917

T=80 sigma\_quadro=0.001 p=6 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.0141511 1.004819 1.003296 0.9872661 1.275385

0.9944354 1.012885 1.002830 0.9912976 1.275343

0.9984570 1.017794 1.010552 0.9879296 1.261413

0.9814476 1.024539 1.007225 0.9889303 1.271158

0.9989512 1.019912 1.015331 0.9931185 1.259385

0.9897814 1.019347 1.022833 0.9912048 1.237746

0.9938549 1.007414 1.032968 0.9925810 1.197704

0.9892852 1.017329 1.041948 0.9963574 1.169202

0.9892852 1.017329 1.041948 0.9963574 1.169202

0.9999106 1.011880 1.028506 0.9966071 1.148557

nu=0.01, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

0.995724 1.017659 1.059085 1.277476 0.9639937

1.005052 1.044390 1.101682 1.257762 0.9093299

1.009630 1.056016 1.099667 1.239561 0.9003058

1.036447 1.055429 1.104006 1.221009 0.9014157

1.009908 1.063189 1.103832 1.204028 0.9031415

1.039504 1.061504 1.103244 1.191078 0.9062476

1.014668 1.063607 1.101736 1.176583 0.9047168

1.026965 1.059785 1.101568 1.162882 0.8831195

1.026965 1.059785 1.101568 1.162882 0.8831195

1.021089 1.058055 1.095062 1.138061 0.8586574

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.000097 1.031436 1.174172 1.266829 1.075714

1.051070 1.093563 1.186225 1.233233 1.069330

1.035916 1.089842 1.188024 1.206494 1.064963

1.063274 1.108321 1.179915 1.182396 1.070145

1.043568 1.099125 1.170910 1.161798 1.079452

1.054771 1.096634 1.160601 1.141037 1.086154

1.056346 1.096598 1.151258 1.123846 1.095271

1.060821 1.093151 1.144916 1.108499 1.095510

1.060821 1.093151 1.144916 1.108499 1.095510

1.051679 1.081399 1.130571 1.084016 1.073630

T=80 sigma\_quadro=0.0001 p=6 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.0071763 0.9777742 1.0097749 0.9897992 1.058603

1.0065618 0.9912102 1.0019993 0.9892795 1.049372

1.0084469 0.9897257 1.0025986 0.9917708 1.048112

0.9910016 1.0011768 1.0018068 0.9863834 1.045054

1.0087238 0.9859350 1.0009709 0.9840325 1.039635

0.9895576 1.0006887 0.9998065 0.9778285 1.036548

1.0081600 1.0004382 1.0074522 0.9760590 1.033899

0.9885961 1.0003633 1.0040175 0.9771578 1.032145

0.9885961 1.0003633 1.0040175 0.9771578 1.032145

0.9946404 1.0072683 1.0004799 0.9741732 1.028670

nu=0.01, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.007079 1.034253 1.133814 1.276716 1.086644

1.015355 1.060403 1.126242 1.255904 1.033750

1.019223 1.067296 1.119229 1.235171 1.023090

1.022034 1.062181 1.115542 1.216289 1.021238

1.020147 1.060017 1.110545 1.200542 1.020900

1.022247 1.058991 1.106737 1.186336 1.021135

1.022420 1.056442 1.101890 1.171876 1.021621

1.021181 1.050636 1.097411 1.159053 1.022172

1.021181 1.050636 1.097411 1.159053 1.022172

1.025618 1.045616 1.089695 1.136606 1.023802



nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.038996 1.100796 1.215818 1.283830 0.9834639  
1.077106 1.113015 1.203653 1.247422 0.9884102  
1.064018 1.109163 1.191131 1.215955 0.9879329  
1.072935 1.103559 1.181404 1.189272 0.9874429  
1.066130 1.099810 1.172996 1.165724 0.9872402  
1.065180 1.094291 1.164348 1.145475 0.9873968  
1.061377 1.091195 1.155593 1.127725 0.9876777  
1.058387 1.084043 1.147061 1.112462 0.9878425  
1.058387 1.084043 1.147061 1.112462 0.9878425  
1.052516 1.078283 1.132115 1.087465 0.9882013

## Esperimento 2: MSE ottimali / MSE lineari

T=40 sigma\_quadro=0.001 p=4 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

0.9695345 1.0065321 0.9981364 0.9937093 0.9301779  
0.9993281 1.0069567 1.0067711 0.9852507 0.9476029  
0.9817634 1.0263293 1.0014849 0.9752112 0.9569766  
1.0133759 1.0125418 1.0040301 0.9755676 0.9616764  
0.9881272 1.0058622 0.9928257 0.9798554 0.9675262  
0.9952969 1.0036379 1.0026875 0.9807049 0.9715967  
0.9837104 1.0067017 1.0021189 0.9843994 0.9718575  
0.9842842 0.9991659 0.9992099 0.9895162 0.9756685  
0.9842842 0.9991659 0.9992099 0.9895162 0.9756685  
0.9999415 0.9955112 0.9966967 0.9945544 0.9764036

T=40 sigma\_quadro=0.0001 p=4 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.0142435 0.9974593 1.0086625 0.9988340 0.9852038  
1.0092334 1.0036780 0.9899915 1.0081161 0.9843407  
1.0168018 1.0257537 0.9885810 1.0029521 0.9840778  
0.9977568 1.0142886 0.9983489 1.0014770 0.9828271  
1.0012326 1.0081207 1.0085974 0.9979517 0.9830777  
1.0062430 1.0134188 1.0120324 1.0020543 0.9845807

1.0004989 1.0193570 1.0085625 0.9975283 0.9856121  
1.0160551 1.0122815 1.0084599 0.9984824 0.9864893  
1.0160551 1.0122815 1.0084599 0.9984824 0.9864893  
1.0287482 1.0196549 1.0088664 1.0072329 0.9871982

T=80 sigma\_quadro=0.001 p=6 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

0.9970376 1.0091057 1.0101531 0.9893766 1.118692  
1.0019177 1.0129145 1.0148197 0.9930806 1.150809  
0.9891955 1.0133048 1.0216690 0.9848088 1.145845  
0.9931632 0.9973267 1.0250551 0.9895098 1.164992  
0.9939398 1.0059225 1.0246600 0.9928392 1.179819  
1.0072803 1.0022105 1.0189782 0.9949688 1.191447  
0.9934887 1.0014404 1.0078694 0.9891939 1.187372  
1.0043753 1.0014180 1.0082671 0.9864327 1.188946  
1.0043753 1.0014180 1.0082671 0.9864327 1.188946  
1.0105998 1.0083819 0.9946701 0.9822075 1.157187

T=80 sigma\_quadro=0.0001 p=6 n=10000

nu=0, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

0.9943988 1.0026994 1.0054158 0.9948580 0.9699281  
0.9749658 0.9815666 0.9913659 0.9979721 0.9726482  
0.9902552 0.9881704 0.9806921 1.0020738 0.9731714  
0.9797656 0.9907809 0.9945345 1.0035385 0.9734195  
0.9844125 0.9835770 0.9879934 1.0091045 0.9719454  
0.9965365 0.9834473 0.9879101 1.0100286 0.9716825  
0.9848241 0.9849324 0.9890153 1.0062848 0.9714264  
0.9975216 0.9991656 0.9851631 1.0040561 0.9714302  
0.9975216 0.9991656 0.9851631 1.0040561 0.9714302  
0.9987307 0.9872994 0.9873543 1.0081555 0.9737594

### Esperimento 3: MSE "Naive" / MSE lineari

T=40 sigma\_quadro=0.001 p=4 n=10000

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.015595 1.020952 1.106762 1.370720 3.194889

1.023409 1.035216 1.113498 1.339195 2.806823

1.022300 1.043799 1.102575 1.310837 2.514772

1.030716 1.045432 1.097602 1.284888 2.289048

1.025570 1.041094 1.092062 1.261507 2.110501

1.032507 1.044116 1.083717 1.240973 1.966692

1.026274 1.039356 1.079630 1.223354 1.848994

1.021272 1.036885 1.075954 1.207378 1.751382

1.021272 1.036885 1.075954 1.207378 1.751382

1.033059 1.028897 1.069360 1.178935 1.600156

nu=0.05, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.062543 1.131038 1.264222 1.690770 4.045009

1.089768 1.121824 1.242620 1.621426 3.381938

1.063300 1.118465 1.223523 1.561513 2.920407

1.072651 1.109127 1.208360 1.509623 2.584678

1.055051 1.099520 1.191340 1.463971 2.331840

1.059076 1.094869 1.176241 1.423824 2.136123

1.049892 1.088179 1.164499 1.388459 1.981148

1.054058 1.080693 1.153295 1.356994 1.856145

1.054058 1.080693 1.153295 1.356994 1.856145

1.044343 1.072580 1.133498 1.304231 1.668854

T=40 sigma\_quadro=0.0001 p=4 n=10000

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.007235 1.050909 1.125806 1.376310 3.195609

1.030073 1.054839 1.118941 1.344535 2.807624

1.026831 1.053978 1.111350 1.316642 2.515528

1.028204 1.048945 1.103142 1.291861 2.289634

1.024936 1.044869 1.095710 1.269681 2.111025

1.026510 1.040601 1.088515 1.249539 1.967156

1.020723 1.039597 1.082909 1.231327 1.849426

1.019884 1.036463 1.077757 1.214788 1.751801

1.019884 1.036463 1.077757 1.214788 1.751801

1.017579 1.033415 1.067981 1.186023 1.600584

nu=0.05, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.079034 1.139553 1.266202 1.691539 4.044949

1.085432 1.129195 1.244601 1.622574 3.381864

1.077582 1.119827 1.225219 1.562840 2.920356

1.074153 1.109321 1.208234 1.510834 2.584661

1.066701 1.102546 1.193106 1.465329 2.331855

1.063674 1.095053 1.178983 1.425277 2.136121

1.057243 1.089019 1.166302 1.389898 1.981126

1.054855 1.083674 1.155262 1.358475 1.856125

1.054855 1.083674 1.155262 1.358475 1.856125

1.047604 1.073023 1.135320 1.305375 1.668852

T=80 sigma\_quadro=0.001 p=6 n=10000

nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9

1.019494 1.050373 1.217255 1.637908 4.211117

1.052918 1.109372 1.231918 1.601651 3.824542

1.040605 1.102826 1.223573 1.572416 3.506032

1.067311 1.107211 1.214052 1.543004 3.240370

1.060118 1.104839 1.204928 1.516865 3.016148

1.066108 1.104584 1.194017 1.494037 2.824931

1.063164 1.099572 1.186691 1.471490 2.660374

1.060636 1.096722 1.180573 1.449919 2.517586

1.060636 1.096722 1.180573 1.449919 2.517586

1.052023 1.088499 1.166448 1.410663 2.282947  
nu=0.05, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9  
1.095177 1.240807 1.463427 2.096799 4.685467  
1.161785 1.239382 1.441309 2.030168 3.861038  
1.140986 1.237247 1.420319 1.968101 3.487011  
1.147029 1.225846 1.402461 1.911038 3.200604  
1.135222 1.220229 1.386314 1.858244 2.973186  
1.135163 1.210304 1.369772 1.809750 2.784223  
1.123314 1.200575 1.353600 1.765286 2.623949  
1.122693 1.193042 1.338510 1.723969 2.485979  
1.122693 1.193042 1.338510 1.723969 2.485979  
1.114172 1.179703 1.311730 1.650955 2.260471

T=80 sigma\_quadro=0.0001 p=6 n=10000  
nu=0.02, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9  
1.049199 1.114230 1.235230 1.631933 4.210976  
1.073914 1.118095 1.222883 1.601226 3.824541  
1.069065 1.109290 1.214121 1.571850 3.506192  
1.070591 1.109270 1.205218 1.544532 3.240522  
1.066026 1.102835 1.197598 1.519189 3.016229  
1.063849 1.099208 1.191729 1.495021 2.824951  
1.064148 1.094614 1.184270 1.472269 2.660345  
1.059846 1.090717 1.177884 1.451202 2.517538  
1.059846 1.090717 1.177884 1.451202 2.517538  
1.055343 1.084832 1.165447 1.412801 2.282952

nu=0.05, rho da sinistra a destra = -0.9, -0.5, 0, 0.5, 0.9  
1.160252 1.251213 1.459305 2.097815 4.686658  
1.168881 1.245087 1.438749 2.029652 3.861854  
1.163796 1.232882 1.419767 1.967332 3.487706  
1.156747 1.223735 1.401663 1.910326 3.201191  
1.150760 1.215132 1.384805 1.857865 2.973698  
1.143862 1.206622 1.368364 1.809679 2.784673  
1.139092 1.199225 1.352673 1.765289 2.624359  
1.133530 1.192077 1.337837 1.724350 2.486345  
1.133530 1.192077 1.337837 1.724350 2.486345  
1.124160 1.177840 1.310816 1.651174 2.260766



## 10. Bibliografia

- LUTKEPOHL H., XU F., (2009), *The Role of the Log Transformations in Forecasting Economic Variables*, articolo scaricabile ai link: [www.SSRN.com](http://www.SSRN.com), [www.RePEc.org](http://www.RePEc.org), [www.CESifo-group.org/wp](http://www.CESifo-group.org/wp).
- DI FONZO T., LISI F., (2005), *Serie Storiche Economiche*, Carrocci, Roma.
- GUERRERO V. M., (1993) *Time-serie Analysis supported by Power Transformations*, articolo in: *Journal of Forecasting*, Vol.12, 37 – 48 (1993).
- CHATFIELD C., (1996), *The Analysis of Time Series, An Introduction*, Chapman & Hall/CRC, Boca Raton, London, New York, Washington D.C.
- SHUMWAY R. H., STOFFER D. S., (2006), *Time Series Analysis and Its Applications*, Springer, New York.
- CRIVELLARI F., (2006), *Analisi Statistica dei dati con R*, Apogeo, Milano.