

Paolo Mezzera

Ottobre 2003

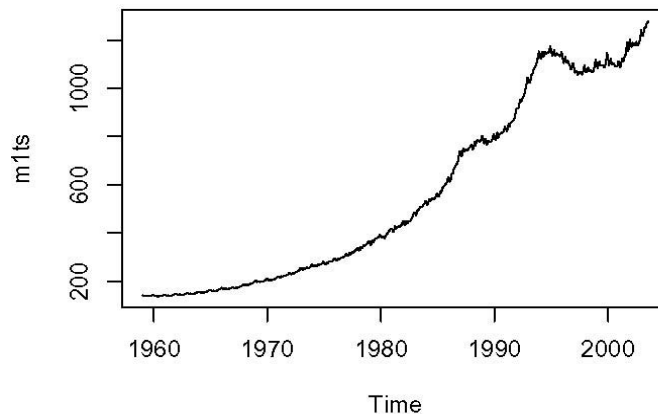
## CONFRONTO TRA TECNICHE DI PREVISIONE: Processi regressivi e reti neurali artificiali

Questo lavoro ha lo scopo di confrontare previsioni ottenute mediante processi regressivi e previsioni ottenute utilizzando le reti neurali artificiali. Entrambi i metodi saranno applicati all'analisi di una serie di tempo, l'offerta di moneta (M1) negli U.S.A. dal Gennaio 1959 all'Agosto 2003.

Sono disponibili 536 dati, su base mensile: dal Gennaio 1959 all'Agosto 2003. Nella figura che segue si può vedere il grafico della serie storica, che ha un andamento indiscutibilmente crescente fino alla seconda metà degli anni '80, per poi continuare a crescere, ma con andamento meno regolare.

Per ottenere il grafico sottostante si usa il programma Open-Source chiamato R. I comandi necessari per ottenere l'immagine del grafico sono:

```
library(ts)
m1 <-read.table(file="c:/m1stock.txt",sep="\t")
mlts<-ts(m1[,3],frequency=12,start=1959)
ts.plot(mlts)
```



Lo scopo di questo lavoro è confrontare due differenti metodi di previsione: processi regressivi e reti neurali artificiali. Le previsioni fatte saranno confrontate tra loro e con la serie originale. Sono stati scelti due intervalli per applicare i metodi e verificare l'abilità dei metodi previsivi: il primo intervallo inizia nel punto in cui la serie perde la sua regolarità (dato 337, Gennaio 1987) e prosegue per 4 anni (dato 384, Dicembre 1990). Il secondo intervallo corrisponde agli ultimi 46 dati della serie (dal dato 490, Ottobre 1999 al dato 536, Agosto 2003).

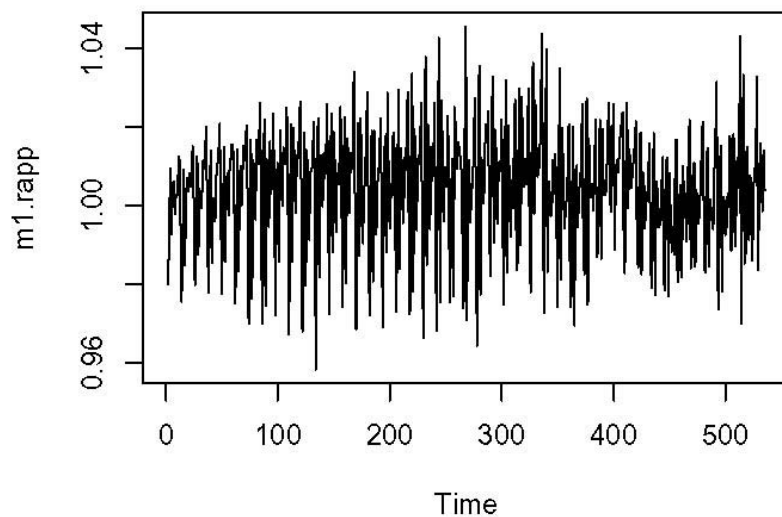
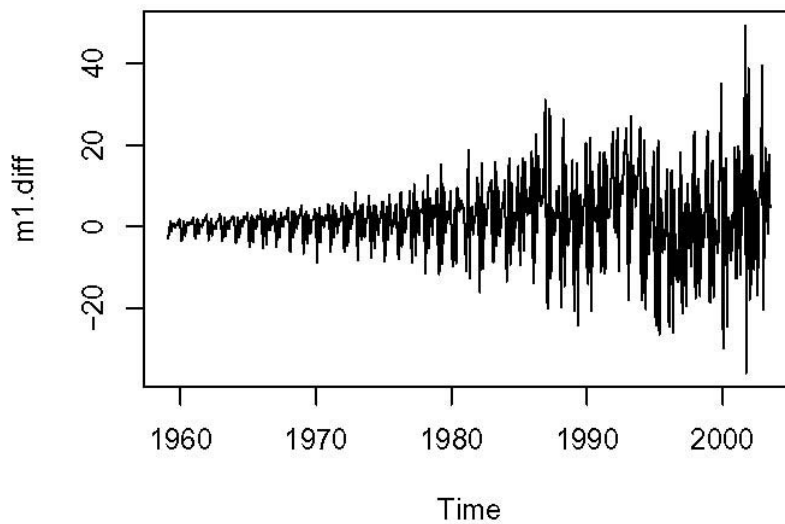
Le fasi principali del lavoro, per ogni previsione, sono le seguenti:

- Valutazione statistica della serie allo scopo di individuare il processo regressivo alla base (AR, MA o ARMA). In contemporanea, si cercherà anche di determinare la stabilità della serie in media e in varianza. Una serie storica che non è stazionaria in media e in varianza generalmente non consente di ottenere buone previsioni. Nel caso la serie non fosse stazionaria in media e varianza, occorrerebbe trasformarla in differenze o, ancora meglio, in rapporti.
- La seconda fase del lavoro consisterà nell'utilizzare i processi AR, MA o ARMA per effettuare previsioni dinamiche utilizzando il programma R.
- Mediante l'utilizzo del programma RNA si procederà con le previsioni statiche.
- Successivamente si procederà con l'analisi della serie tramite il programma RNA DIN per ottenere le previsioni dinamiche.
- In conclusione, mediante l'ausilio di grafici, saranno comparati i risultati derivanti dalle varie tecniche utilizzate per confrontare e individuare eventuali affinità e caratteristiche comuni ai diversi modelli

Per ottenere buone previsioni è necessario che la serie utilizzata sia stazionaria in media e in varianza. Come è facilmente intuibile osservando il grafico della pagina precedente la serie M1 non soddisfa questi requisiti. E' dunque opportuno trasformare la serie in differenze e rapporti: l'analisi delle serie tramite differenze o rapporti riduce notevolmente il problema della stazionarietà in media in quanto, esprimendo tutti i valori in relazione uno con l'altro, viene automaticamente compattato il range dei valori. Maggiori dubbi sull'efficacia di questa tecnica sorgono per quanto riguarda la stazionarietà in varianza. Certamente gli effetti che subisce rispetto alla serie originale sono positivi, ma l'effetto che si ottiene su questo parametro normalmente è inferiore a quello che si ottiene sulla media.

Per ottenere le nuove serie (chiamate `m1.diff` e `m1.rapp`) e per rappresentarle graficamente, sono necessari i seguenti comandi di R:

```
m1.diff<-diff(m1ts, lag=1, differences=1)
m1.rapp<-rep(0,536)
for(i in 2:536) m1.rapp[i]<-m1ts[i]/m1ts[i-1]
m1.rapp[1]<-NA
par(mfrow=c(2,1))
ts.plot(m1.diff)
ts.plot(m1.rapp)
```



Come si può osservare le due serie non risultano perfettamente stazionarie né in media, né in varianza. La serie `m1.diff` appare stazionaria in media, ma assolutamente non in varianza; invece la serie `m1.rapp` non appare perfettamente stazionaria né in media, né in varianza, ma tuttavia sembra sufficientemente stazionaria per essere considerata una buona base di partenza per ottenere delle buone previsioni. Ai fini di questa trattazione, ci accontentiamo di assumere le serie così trasformate sufficientemente stazionarie tali da consentire l'implementazione delle relative tecniche di previsione. Le previsioni, tuttavia, saranno effettuate utilizzando la serie `m1.rapp` che può essere considerata quella con stazionarietà (in media e varianza) maggiore tra le due, o, se si preferisce, la "meno peggio" in quanto a mancanza di stazionarietà.

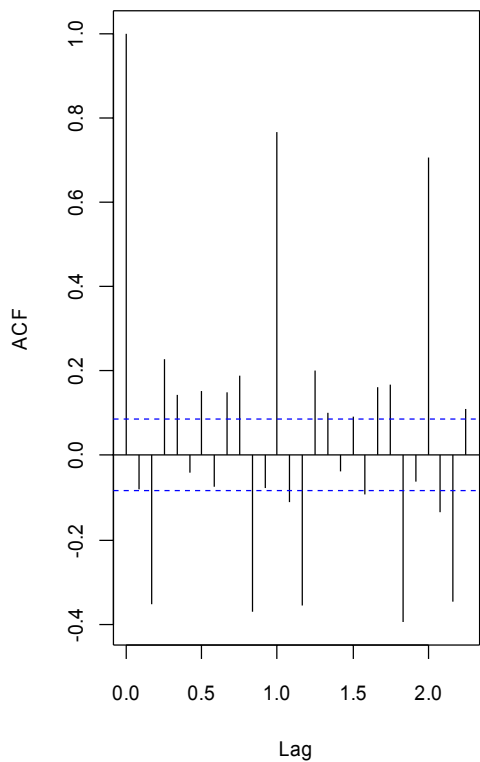
Prima di procedere con le previsioni è necessario scegliere il tipo di processo regressivo da utilizzare. Individuare il tipo di processo regressivo alla base della serie è molto importante ai fini di una buona trattazione ed analisi soprattutto per quanto riguarda l'impiego dei processi AR, MA e ARMA; se si dovesse scegliere erroneamente il tipo di processo, si correrebbe il rischio di ottenere delle previsioni poco precise e conseguentemente poco utili. Per individuare il tipo di processo regressivo alla base delle nostre serie, saranno utilizzati due strumenti statistici che prendono il nome di autocorrelazioni totali e parziali. L'involuppo di questi due parametri dovrebbe consentire di individuare in modo piuttosto univoco, il tipo di processo regressivo alla base delle serie.

Per ottenere le autocorrelazioni totali e parziali sono necessari i seguenti comandi di R:

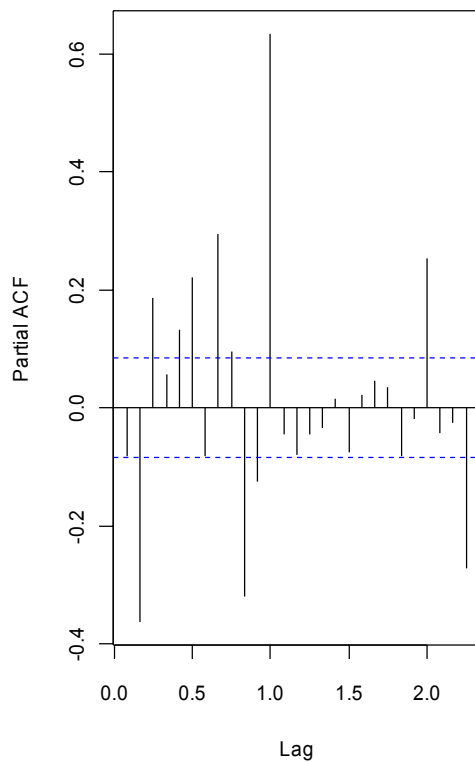
```
par(mfrow=c(1,2))
acf(m1.diff,na.action=na.omit,type="cor")
acf(m1.diff,na.action=na.omit,type="par")
```

```
par(mfrow=c(1,2))
acf(m1.rapp,na.action=na.omit,type="cor")
acf(m1.rapp,na.action=na.omit,type="par")
```

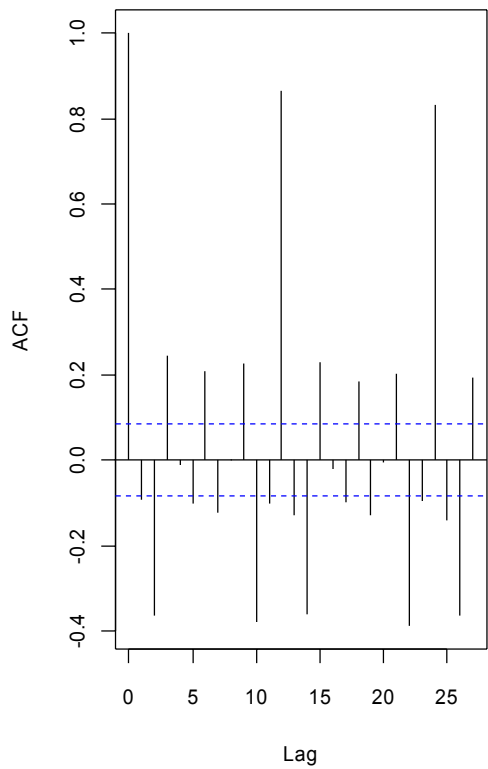
**Series m1.diff**



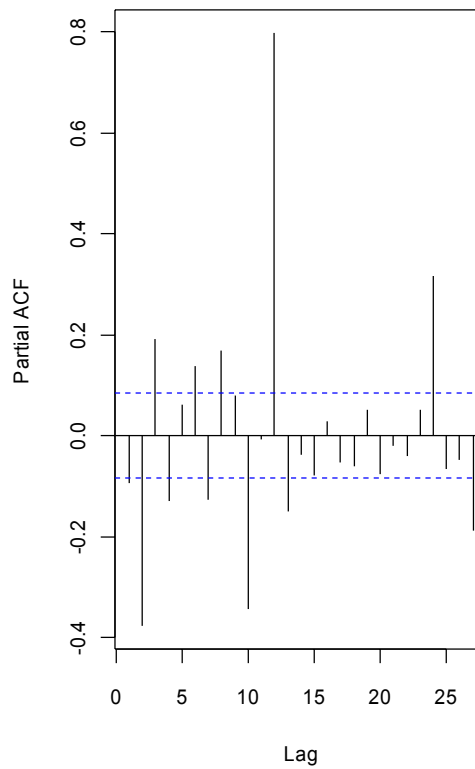
**Series m1.diff**



**Series m1.rapp**



**Series m1.rapp**



Osservando i grafici con le autocorrelazioni non è possibile ricondurre la serie ottenuta a nessun particolare modello. Generalmente si può riconoscere un'autoregressione, una media mobile, o un ARMA, osservando i grafici delle autocorrelazioni totali e parziali, ma in questo caso non si presenta una situazione immediatamente riconoscibile. Non potendo affidare tutte le scelte ad uno schema predefinito è stato necessario fare alcune prove per verificare quale fosse il metodo migliore. Il paragrafo che segue spiega che modello sarà utilizzato per le previsioni.

Per ottenere le previsioni sarà utilizzato il modello ARIMA (AutoRegressivo Integrato a Media Mobile), i modelli ARIMA sono considerati la categoria più generale dei modelli per previsioni di serie storiche che possono essere stationarizzate dalle trasformazioni in differenze e rapporti. Un modello ARIMA è definito da tre parametri: p (l'ordine di AR), d (il grado della differenza) e q (l'ordine di MA): ARIMA(p,d,q).

Alcuni esempi:

ARIMA(0,0,0)=WN(0, $\sigma_\varepsilon^2$ );

ARIMA(1,0,0)=AR(1);

ARIMA(1,0,1)=ARMA(1,1);

ARIMA(0,1,0)=Random Walk

Grazie al modello ARIMA è possibile ottenere i più importanti tipi di processi regressivi (AR, MA, ARMA) semplicemente impostando i valori dei tre parametri p, d e q.

Le previsioni in questo lavoro saranno fatte tutte utilizzando ARIMA considerando solamente il processo autoregressivo (dunque:  $p > 0$ ,  $d = q = 0$ ). Per conoscere l'ordine di AR che è opportuno usare è sufficiente dare i seguenti comandi a R (esempio utilizzato per scoprire l'ordine della serie `m1.rapp`, cioè quella utilizzata per la previsione sul secondo intervallo):

```
m1.rapp.ar <- ar(m1.rapp[2:490])  
m1.rapp.ar
```

R fornisce l'ordine scelto: `Order selected 25`. Per ottenere buone previsioni si dovrà dunque usare il modello ARIMA(25,0,0): questo metodo conduce ad ottenere buoni risultati, come si vedrà più avanti.

Le previsioni, come già spiegato prima, si faranno solo sulla serie `m1.rapp`, e non sulla serie originale, né sulla serie `m1.diff`.

Per ottenere le previsioni, utilizzando R, sono necessari i seguenti comandi:

```
m1.rapp.s.arima<- arima(m1.rapp[2:337],order=c(24,0,0))
m1.rapp.s.arima.predict<-predict.Arima(m1.rapp.arima,n.ahead=47)
```

Si ottiene così la previsione per l'intervallo Gennaio 1987 – Dicembre 1990. Il motivo per cui è stato impostato `order=c(24,0,0)` è appunto quello spiegato poche righe sopra. La previsione ottenuta sarà confrontata con quelle che saranno ottenute tramite il metodo delle reti neurali artificiali (previsioni statiche e dinamiche utilizzando il programma RNA).

I comandi necessari per ottenere le previsioni per il secondo intervallo (Ottobre 1999 - Agosto 2003) sono:

```
m1.rapp.arima <- arima(m1.rapp[2:490],order=c(25,0,0))
m1.rapp.arima.predict <- predict.Arima(m1.rapp.arima,n.ahead=46)
```

Per ottenere le previsioni statiche sugli stessi intervalli utilizzando il metodo delle reti neurali artificiali occorre:

- Esportare da R la serie `m1.rapp` in un file
- Determinare la dimensione della finestra comprensiva dei valori di input e dei valori di output con relativa creazione del file (utilizzando il programma `FINESTRA`)
- Impostare il file appena creato per il relativo utilizzo in RNA mediante l'inserimento delle informazioni necessarie al programma per effettuare le previsioni tenendo debito conto di lasciare gli ultimi pattern a scopo di verifica
- Esecuzione del programma RNA per effettuare l'apprendimento mediante la creazione dei file contenenti i pesi
- Modifica del file campione per effettuare le previsioni solo sugli ultimi valori precedentemente scartati facendo uso dei pesi appena calcolati.

Per ottenere le previsioni dinamiche:

- Impostare dei nuovi file RNA che consentano di effettuare previsioni dinamiche e forniscano informazioni al programma RNA DIN (versione di RNA apposta per previsioni dinamiche). Le previsioni effettuate in modo dinamico prevedono i valori di periodo in periodo e le previsioni per i periodi successivi si basano sulle previsioni dei periodi precedenti anch'esse già stimate.

- Impostare il file DIN, che deve contenere informazioni aggiuntive necessarie al programma per effettuare previsioni dinamiche
- Esecuzione del programma RNA per effettuare l'apprendimento mediante la creazione dei file contenenti i pesi
- Modifica del file ed esecuzione del programma RNA DIN per ottenere le previsioni dinamiche

Il primo passo consiste nell'esportazione della serie `m1.rapp` da R:

```
write(m1.rapp, file="c:\m1.rapp.txt", ncol=1)
```

Con il programma `FINESTRA` è necessario creare un file che riproduca i valori contenuti in `m1.rapp.txt` in forma di matrice. La serie originale è composta di 536 dati, la serie `m1.rapp`, invece, è composta da 535 dati. Le dimensioni della matrice saranno: 13 (12+1) x 523.

Fino a questo punto non c'è alcuna differenza tra ciò che occorre fare per ottenere la previsione sul primo intervallo e sul secondo. Il file creato con `FINESTRA` è valido per entrambi. Per quanto riguarda, invece, i successivi files, essi dovranno essere necessariamente diversi.

Il file RNA necessario per fare l'apprendimento, per la previsione del primo intervallo, sarà il seguente:

```
523 324 12 9 1
[Pattern (matrice 13x523)]
12345
52300 s i
0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s
```

per il secondo intervallo:

```
523 477 12 9 1
[Pattern (matrice 13x523)]
12345
52300 s i
```



```

0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s

```

Come si può notare, l'unica differenza tra i due file si trova al secondo posto della prima riga. E' il numero di pattern da usare in sede di apprendimento, e chiaramente devono essere diversi. Gli altri valori: 523 è il numero di pattern totali, quindi le righe della matrice creata con FINESTRA dalla serie composta da 535 dati; 12 sono i nodi di input; 9 i nodi hidden (nascosti); 1 è il nodo di output. I nodi nascosti sono pari a 9 perché è un numero intermedio tra 12 e 1 e per i buoni risultati mostrati da alcune prove, in cui sono stati usati anche altri numeri. E' possibile, con ulteriori tentativi, giungere ad ottenere risultati migliori di quelli che sono stati ottenuti, e che saranno presentati fra poco, grazie ad un diverso numero di nodi hidden. 12345 è l'innesco dei numeri casuali (fissato pari a 12345 per una specie di convenzione; in realtà si può inserire qualsiasi numero compreso tra 0 e 65535). 52300 è il numero di cicli dell'apprendimento, fissato pari ad un multiplo del numero di pattern. Un'altra impostazione scelta: la s dopo 52300 significa che i dati saranno analizzati in forma sequenziale (e non casuale).

Con i files descritti si svolge l'apprendimento, per ottenere le previsioni statiche con i pesi appena individuati sui pattern rimasti occorre modificare i files:

per le previsioni relative al primo intervallo:

```

523 -324 12 9 1
[Pattern (matrice 13x523)]
0
0 s i
0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s

```

per le previsioni relative al secondo intervallo:

```

523 -477 12 9 1
[Pattern (matrice 13x523)]
0
0 s i
0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s

```

Data l'attuale impostazione dei file, RNA scarcerà i primi  $n$  dati per eseguire la valutazione soltanto su quelli residui. Avendo indicato un innesco dei numeri casuali pari a 0, verranno utilizzati come pesi quelli indicati nei file con estensione .1 e .2 ed avendo fissato il numero di cicli a zero, non verrà effettuato alcun apprendimento ma, dati i valori in input, verranno calcolati semplicemente i relativi output che rappresenteranno le previsioni desiderate.

Per ottenere le previsioni dinamiche con RNA bisogna usare l'apposita versione RNA DIN, e occorre impostare dei nuovi file RNA che consentano di effettuare previsioni dinamiche e forniscano informazioni allo stesso RNA DIN.

I files con estensione .rna necessari per l'apprendimento saranno i seguenti:

per il primo intervallo:

```

523 324 12 9 1
[Pattern (matrice 13x523)]
12345
52300 s i
0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s

```

per il secondo intervallo:

```

523 477 12 9 1
[Pattern (matrice 13x523]
12345
52300 s i
0.8
0.3
[nome files di output]
0.6 0.9
0.6 0.9
n n s

```

Accanto ai file con estensione .rna, l'utilizzo di RNA DIN richiede l'implementazione di un altro file con estensione .din per ciascun file .rna presente. All'interno di tale file occorre inserire informazioni aggiuntive necessarie al programma per effettuare previsioni dinamiche. I file con estensione .din utilizzati risultano i seguenti:

per le previsioni relative al primo intervallo:

```

324 47 13
1 1 1 1 1 1 1 1 1 1 1 1 1

```

per le previsioni relative al secondo intervallo:

```

477 46 13
1 1 1 1 1 1 1 1 1 1 1 1 1

```

Il primo numero di questi files indica il numero di pattern del file .rna da utilizzare come dati statici. Il secondo indica la previsione da effettuare in avanti, il terzo parametro rappresenta la dimensione della finestra che coinciderà con la somma dei nodi di input e dei nodi di output. Infine occorre specificare un vettore di 0 o 1 relativo ai dati della finestra che si desidera utilizzare: 0 se il relativo dato non lo si vuole utilizzare, 1 se lo si vuole utilizzare per la previsione. In questo caso tutti i dati vengono presi in considerazione in entrambe le prove. Dopo aver impostato i suddetti file è sufficiente eseguire prima RNA affinché esegua l'apprendimento sui pattern desiderati, e successivamente eseguire RNA DIN affinché prelevi le informazioni dai file generati da RNA relativi ai pesi per effettuare le previsioni dinamiche vere e proprie. Le previsioni così ottenute

verranno inserite in un apposito file con estensione .pre, pronte per essere confrontate e rappresentate graficamente.

Ora che tutte le previsioni sono state ottenute non resta che confrontarle. Per farlo occorre importare in R i dati ottenuti con RNA. I comandi per farlo sono:

```
m1.rapp.13s.erp <- read.table(file="c:/rna/rapp13s.erp")
m1.rapp.13s.erp <- m1.rapp.13s.erp[,1]
m1.rapp.13s.erp <- m1.rapp.13s.erp[1:47]
m1.rapp.13s.din <- read.table(file="c:/rna/rapp13ds.pre")
m1.rapp.13.erp <- read.table(file="c:/rna/rapp13.erp")
m1.rapp.13.erp <- m1.rapp.13.erp[,1]
m1.rapp.13.din <- read.table(file="c:/rna/rapp13d.pre")
```

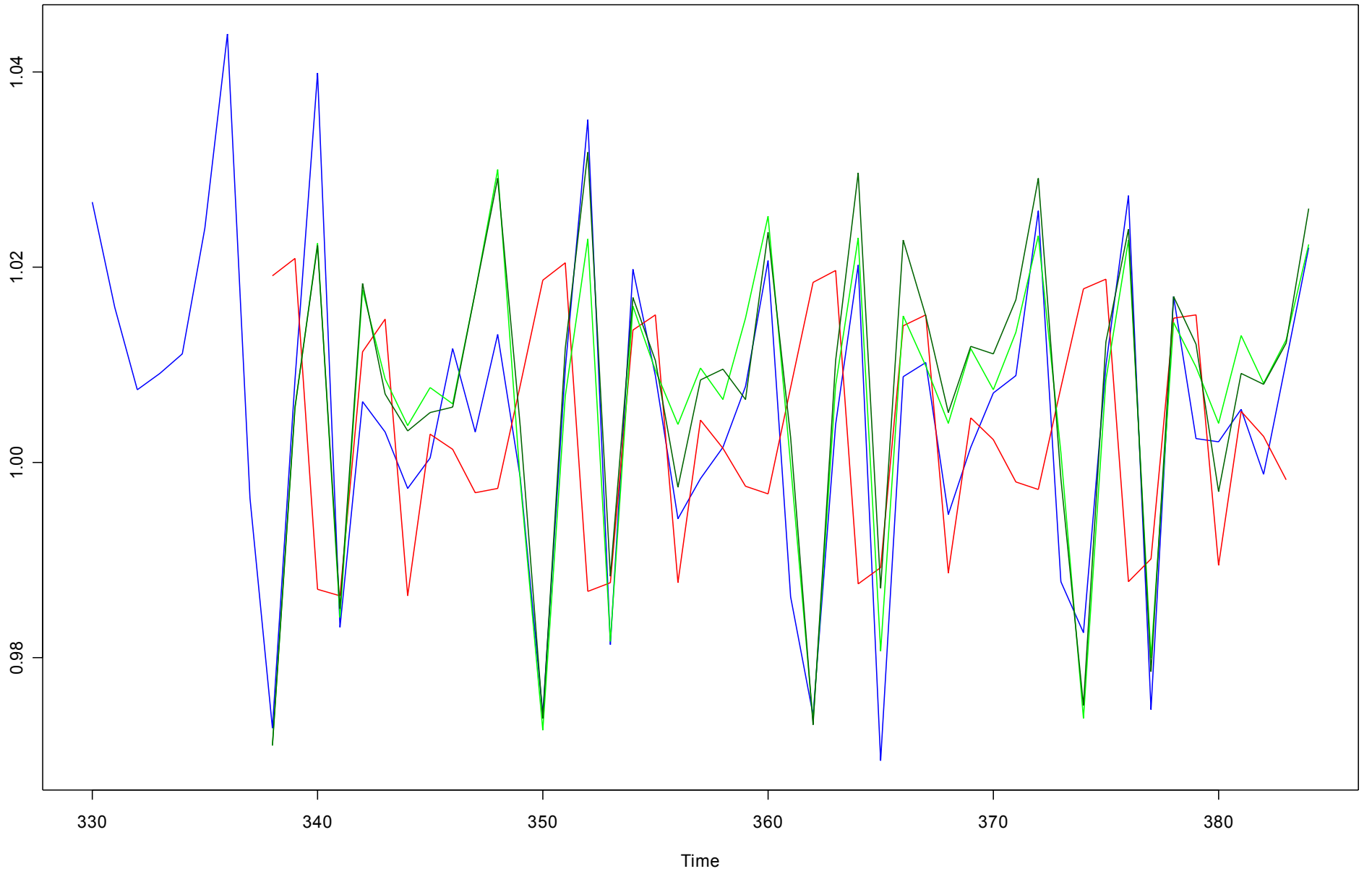
Con R si può procedere alla creazione di due grafici (uno per ogni intervallo oggetto di previsione) in cui poter confrontare l'efficacia dei metodi utilizzati. Per ottenere i grafici occorrono i seguenti comandi:

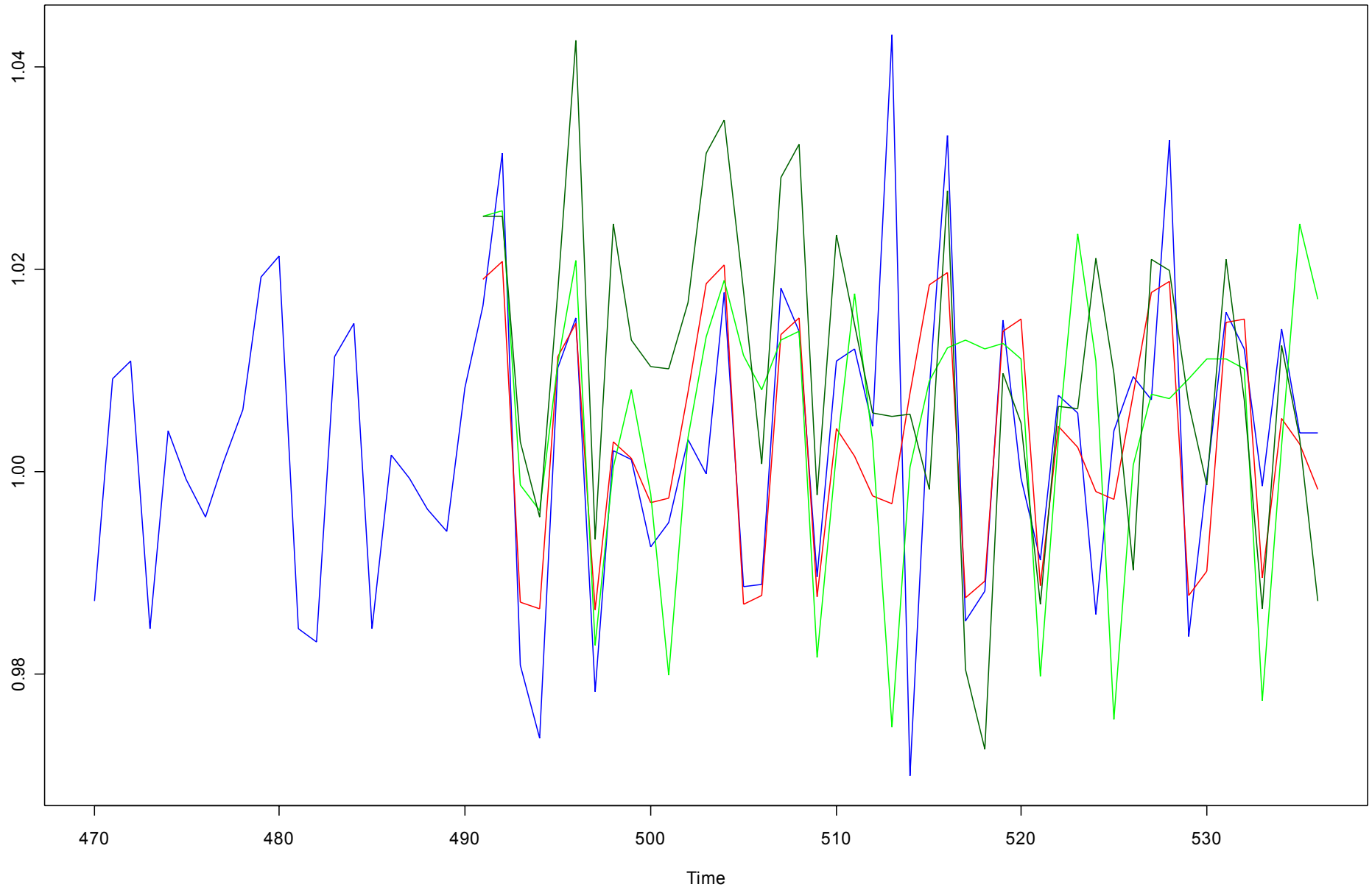
Primo intervallo (serie originale m1.rapp: blu, previsioni con ARIMA: rosso, previsioni statiche con RNA: verde scuro, previsioni dinamiche con RNA: verde chiaro):

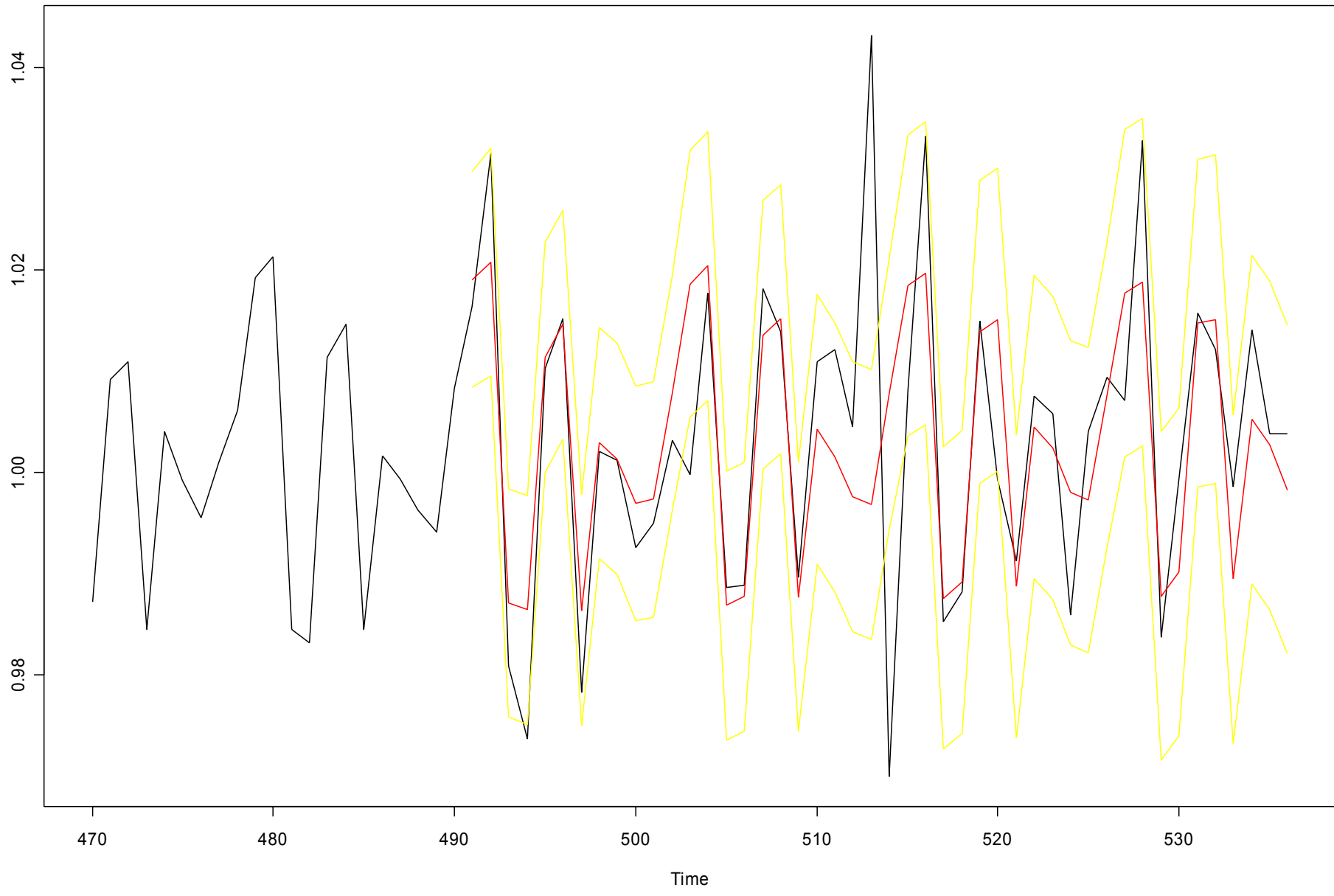
```
ts.plot ( ts(m1.rapp[330:384],start=330),
          ts(m1.rapp.arima.predict$pred,start=338),
          ts(m1.rapp.13s.din,start=338),
          ts(m1.rapp.13s.erp,start=338),
          gpars=list(col=c("blue","red","green","darkgreen"))) )
```

Secondo intervallo (serie originale m1.rapp: blu, previsioni con ARIMA: rosso, previsioni statiche con RNA: verde scuro, previsioni dinamiche con RNA: verde chiaro):

```
ts.plot ( ts(m1.rapp[470:536],start=470),
          ts(m1.rapp.arima.predict$pred,start=491),
          ts(m1.rapp.13.din,start=491),
          ts(m1.rapp.13.erp,start=491),
          gpars=list(col=c("blue","red","green","darkgreen"))) )
```







I grafici così ottenuti visualizzano solamente i dati relativi agli intervalli previsti e pochi valori precedenti della serie `m1.rapp`. Nonostante ciò, e nonostante siano stati scelti colori diversi tra loro, i grafici appaiono illeggibili se non sono visualizzati in grandi dimensioni. Per questo motivo è stata fatta la scelta di stamparli molto grandi e sono stati posizionati nelle due pagine precedenti (per quanto riguarda il terzo grafico si veda più avanti).

Analizzando il primo grafico si nota subito che le due serie previste con le reti neurali artificiali sono quasi identiche: a parte alcuni brevissimi tratti sono sempre sovrapposte. Per quanto riguarda la precisione delle previsioni appare chiaro che il peggior risultato è quello ottenuto tramite il metodo regressivo: spesso l'andamento previsto appare diverso da quello che si è avuto in realtà. Invece, le previsioni effettuate utilizzando le reti neurali artificiali sono più precise: solamente in quattro punti del grafico si ha una vistosa differenza dall'andamento reale. Da notare sono i primi e gli ultimi valori previsti: le previsioni statiche e dinamiche fatte con RNA assumono valori praticamente identici a quelli reali, invece le previsioni fatte con ARIMA forniscono valori sensibilmente diversi da quelli reali.

Il secondo grafico mostra una situazione più confusa: paradossalmente le previsioni statiche con RNA hanno prodotto risultati poco soddisfacenti e sono da ritenere sicuramente inferiori rispetto a quelli prodotti dalle previsioni dinamiche prodotte con RNA e con R. La previsione del metodo regressivo ARIMA è qui molto più precisa rispetto al grafico precedente: non considerando due picchi che non ha assolutamente predetto, l'andamento della serie predetta è molto simile a quello della serie originale, tanto da far considerare questa previsione come la migliore fra le tre. Riguardo alla previsione dinamica ottenuta con l'uso delle reti neurali artificiali essa si pone ad un livello intermedio, seppur non può essere considerata particolarmente migliore della previsione statica.

Per quanto riguarda la tecnica che fa uso di previsioni processi regressivi sarebbe interessante includere nei grafici anche altre due linee che rappresentino l'intervallo di confidenza. Le serie che danno origine alle linee in questione si ottengono sommando (per la prima; sottraendo per la seconda) alla serie predetta gli scarti moltiplicati per 1,96 (o altro valore a scelta; 1,96 è scelto convenzionalmente per gli intervalli di confidenza). Sicuramente appare chiaro che non era conveniente aggiungere ancora due linee ai già densi grafici di cui si è parlato prima. Tuttavia si fornisce qui un esempio:

```
ts.plot ( ts(m1.rapp[470:536],start=470) ,
```



```
ts(m1.rapp.arima.predict$pred,start=491),ts(m1.rapp.arima.predict$pred+1.96*m1.rapp.arima.predict$se,start=491),ts(m1.rapp.arima.predict$pred-1.96*m1.rapp.arima.predict$se,start=491),
gpars=list(col=c("black","red","yellow","yellow")))
```

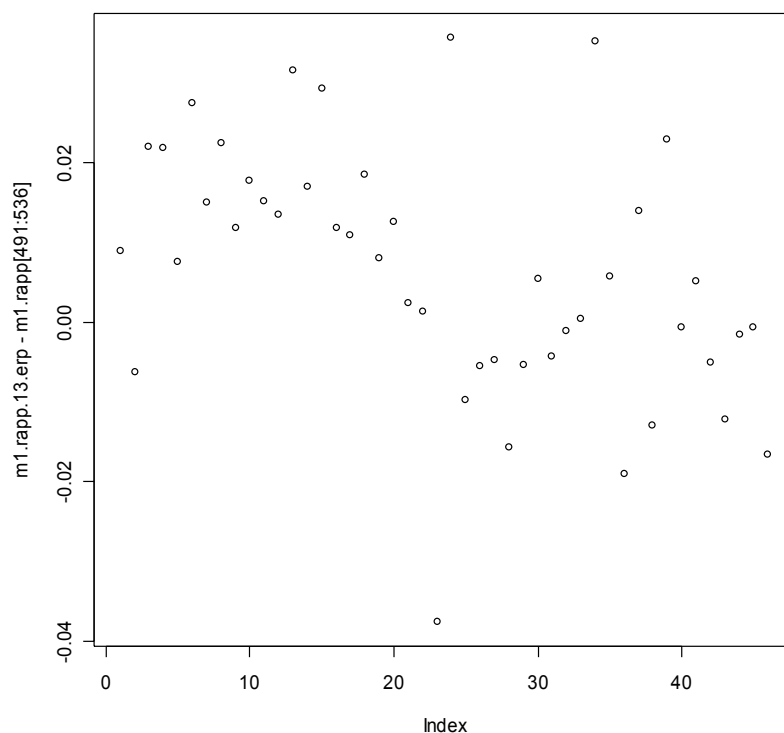
Nel grafico prodotto da questi comandi di R sono presenti tre serie relative al secondo intervallo di previsione: la serie originale in nero, la serie prevista con ARIMA in rosso e le due serie che delimitano l'intervallo di confidenza in giallo.

Osservando il grafico (che si trova a pagina 15) si osserva che il valore reale raramente esce all'infuori del corridoio imposto dalla deviazione standard.

E' possibile poi ottenere, utilizzando R, altri numerosi grafici. Ad esempio, si consideri la previsione statica con RNA sul secondo intervallo: si potrebbero studiare gli scarti ( $m1.rapp$  prevista con RNA -  $m1.rapp$ ) e da essi trarre informazioni per migliorare il metodo previsivo per questa serie. Il comando per ottenere il grafico di questi scarti è:

```
plot(m1.rapp.13.erp-m1.rapp[491:536])
```

e da esso si ottiene il grafico seguente:



In conclusione, se si dovesse definire quale è la tecnica migliore per fare previsioni, certamente sarebbe difficile rispondere. In generale la tecnica probabilmente meno efficiente è quella che fa uso di processi regressivi mentre le tecniche che utilizzano reti neurali sembrano essere caratterizzate da una migliore capacità previsionale. Se dovessimo valutare quale di queste tecniche si è mostrata più efficiente nelle prove effettuate durante questo lavoro forse la “classifica” più onesta porrebbe processi regressivi e previsioni dinamiche con RNA a pari-merito, seguite dalle previsioni statiche con reti neurali artificiali.

Per dare un giudizio serio sulla reale capacità previsiva dei diversi metodi occorre sicuramente fare molte più prove, in particolare provare a variare il numero di nodi (input e hidden, in particolare) delle reti neurali, provare ad usare modelli regressivi diversi o più complicati, ...

---

#### NOTE:

La serie di tempo utilizzata è fornita dal sito web <http://www.economagic.com>

**R** è un software Open-Source, scaricabile all'indirizzo <http://www.r-project.org/>

Distribuito da “The R Foundation”, per informazioni vedere [Free Software Foundation's](#) GNU project

**RNA** (incluse le versioni *rna\_h*, *rna\_r* e *rna\_din*) e **FINESTRA** sono programmi Open-Source scritti per sistemi MS-DOS, disponibili all'indirizzo:

[http://web.econ.unito.it/terna/dinamica\\_tecniche/](http://web.econ.unito.it/terna/dinamica_tecniche/)

Autore dei programmi: Pietro Terna, Dipartimento di Scienze Economiche e Finanziarie " G.Prato", Facoltà di Economia - Università di Torino