

Università degli Studi di Padova



Facoltà di Scienze Statistiche

Corso di Laurea in Statistica Economia e Finanza

**Combinazione di Previsioni per la Volatilità:
applicazioni ad un indice di borsa**

**Combination of Forecasts for the Volatility:
applications to a stock exchange index**

Relatore: Ch.mo Prof. Silvano Bordignon

Laureando: Donadeo Luca - 556075-SEF

Anno Accademico 2008-2009

A mio nonno e a Federico

Indice

Introduzione.....	7
Capitolo 1.....	9
Analisi preliminari del titolo S&P500	
1.1 Dati	9
1.2 Statistiche descrittive dei rendimenti	11
1.3 Test di normalità dei rendimenti	11
1.4 Analisi della dipendenza.....	13
1.5 Volatilità	15
1.5.1 GARCH(1,1).....	16
1.5.2 APARCH(1,1).....	17
Capitolo 2.....	19
Introduzione ai modelli di stima della volatilità	
2.1 MBF.....	19
2.1.1 RV.....	21
2.2 IV-VIX	25
Capitolo 3.....	27
Modelli e combinazioni	
3.1 Come combinare le previsioni.....	27
Capitolo 4.....	33

Valutazione delle previsioni: MSE, QLIKE e Diebold-Mariano Test	
4.1 <i>Singole previsioni</i>	34
4.2 <i>Previsioni combinate</i>	36
Conclusioni	41
Codice programma	43
1. <i>Codice per analisi del capitolo 1</i>	43
2. <i>Codice per previsioni, regressioni per combinazioni e test Diebold-Mariano</i>	45
Bibliografia	60
Ringraziamenti	61

Introduzione

La volatilità è la misura delle fluttuazioni dei rendimenti, essa rappresenta una prima misura del rischio dell'asset considerato ed è anche una misura non osservabile e proprio per questi aspetti negli ultimi anni ha ricevuto un notevole interesse in campo econometrico con lo sviluppo di numerosi modelli intenti a stimarla. Va da sé che riuscire a stimarla risulta quindi importante per chiunque si occupi di investimenti, di valutazioni finanziarie, di politiche monetarie, analisi di portafoglio ecc. Numerosi sono gli studi, come detto, e numerosi sono quelli che si occupano di individuare un modo migliore per riuscire a stimarla.

Intendendo per volatilità la varianza dei rendimenti condizionata all'informazione disponibile al tempo t , $Var(r_t|I_{t-1})$, lo scopo di questo lavoro è quello di studiare e di individuare un modo migliore per prevederla. In particolare, partendo da modelli econometrici e non solo si è andati a cercare quale singola previsione ottenuta o quale combinazione di previsioni generata produce una stima della volatilità futura migliore. Lo spunto è stato preso dallo studio di *Becker e Clements: "Are combination forecasts of S&P 500 volatility statistically superior?" (2008)*.

Per testare quale previsore stima meglio la volatilità futura sono stati utilizzati modelli econometrici semplici, quali il GARCH il TSGARCH e l'APARCH, è stato utilizzato un modello ARMA per la volatilità realizzata dell'indice (RV) ed infine è stato utilizzato anche uno strumento del mercato: la volatilità implicita dello S&P 500, il VIX. Oltre a previsioni

singole sono state studiate anche combinazioni di previsioni, dove i pesi delle combinazioni sono stati stimati tramite regressioni della volatilità realizzata sulle previsioni ottenute dai singoli modelli.

Vogliamo individuare se c'è una combinazione di previsioni che fornisce performance superiori rispetto alle singole previsioni. Per ottenere risultati sui quali prendere decisioni utilizzeremo due strumenti: uno è costituito da due funzioni di perdita: MSE e QLIKE, il secondo riguarda l'utilizzo del test statistico di Diebold-Mariano.

I risultati mostrano che il VIX, dà previsioni di volatilità inferiore rispetto a quelle ottenute da tutti gli altri modelli e rispetto alle loro combinazioni. Ciò indica che, contrariamente a quanto evidenziato da altri studi (Lamoureux e Lastrapes (1993), Fleming, Ostdiek e Whaley (1995) e altri), il VIX non sempre rappresenta una misura attendibile per la stima della volatilità futura.

Il lavoro è così organizzato: nel primo capitolo faremo una breve analisi della serie dello S&P 500 utilizzata; si procederà poi, nel capitolo 2, con un'introduzione ai modelli utilizzati per generare previsioni; nel capitolo 3 si spiegherà il metodo utilizzato per ottenere le previsioni e le combinazioni ed infine nel capitolo 4, dopo una breve introduzione alle misure e statistiche utilizzate per la valutazione, si riportano i risultati empirici.

Capitolo 1

Analisi preliminari del titolo S&P500

1.1 Dati

Questo studio si basa sui dati di chiusura giornalieri dell'indice S&P 500, indice che segue l'andamento di un paniere azionario formato dalle 500 aziende statunitensi a maggiore capitalizzazione; i dati sono stati raccolti per il periodo che va dal 29 febbraio 2000 al 28 febbraio 2005 per un totale di 1251 osservazioni. Dati scaricati dal sito YahooFinance attraverso il programma R.

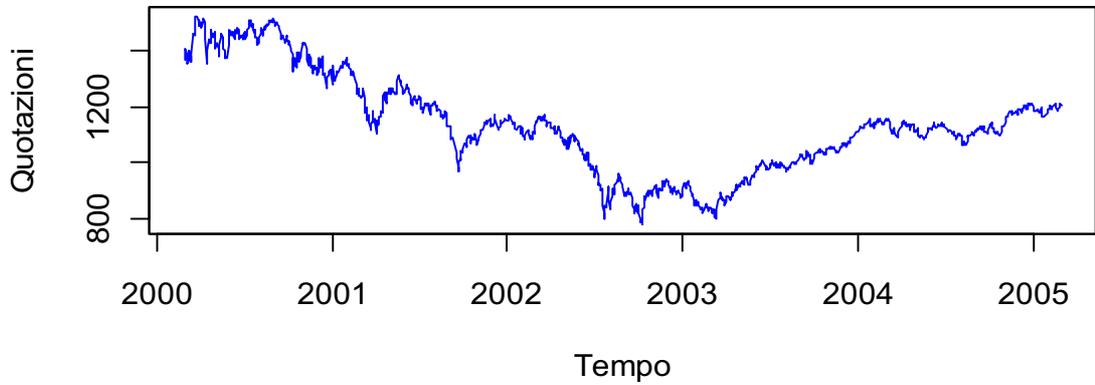
I rendimenti utilizzati sono definiti come la differenza prima del logaritmo della serie dei prezzi in percentuale:

$$r_t = 100 * [\log(p_t) - \log(p_{t-1})]$$

Dove p_t è il prezzo del titolo.

Di seguito sono riportati i grafici della serie dei prezzi e dei rendimenti dell'indice S&P500

Quotazione giornaliera S&P500



Rendimenti logaritmici S&P500

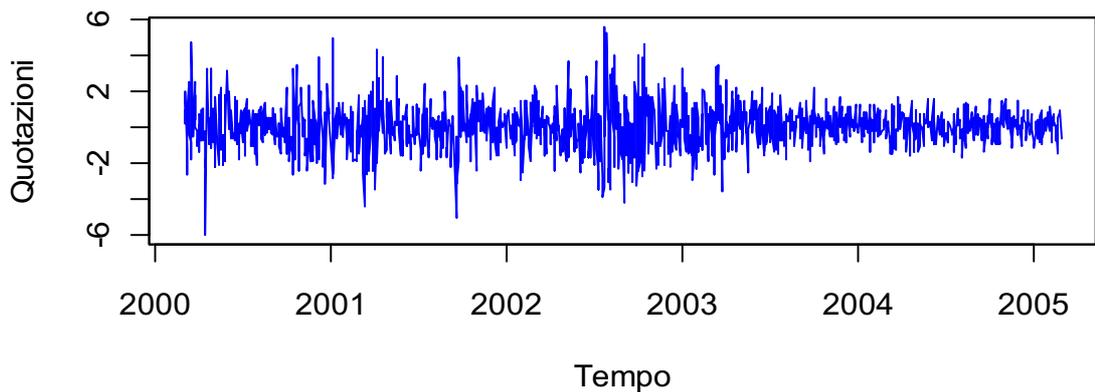


Figure 1.1.1: serie giornaliera dei prezzi dello S&P 500 (sopra) e dei rendimenti (sotto)

Nella figura dei prezzi si può notare una decrescita nel periodo che va dal 2000 al 2003 seguita da una lieve ripresa.

Dal grafico dei rendimenti si può notare il fenomeno del *volatility clustering*, i periodi di alta volatilità si concentrano, stessa cosa per i periodi di bassa volatilità. Inoltre si può notare come nel 2003 la volatilità sia abbastanza instabile e accentuata rispetto al periodo 2004/2005.

1.2 Statistiche descrittive dei rendimenti

	r_t
Mean	-0.004830
Median	0.007202
Sum	-5.994152
SE Mean	0.035165
Variance	1.534603
Stdev	1.238791
Skewness	0.2006900
Kurtosis	1.854083

Table 1.2.1: statistiche descrittive dei rendimenti dello S&P 500

Da queste statistiche possiamo notare un eccesso di curtosi, misura della pesantezza delle code, ben diverso da quello della Normale che risulterebbe pari a zero (la curtosi della normale è 3, l'eccesso 0); si può notare anche una leggera asimmetria positiva. La media sull'ordine dello -0,48% può essere statisticamente considerata pari a zero infatti il test di nullità:

$$|t| = \frac{Mean}{SE\ mean} = 0.1373525$$

dà un valore abbondantemente inferiore al convenzionale 1.96, valore critico al 95%.

1.3 Test di normalità dei rendimenti

Jarque - Bera Normality Test 197.305

p-value JB test < 2.2e-16

Shapiro - Wilk Normality Test 0.9792

P VALUE: 1.906e-12

I due test di normalità dei rendimenti rifiutano l'ipotesi che questi ultimi si distribuiscano come una Normale; nei prossimi due grafici si può notare la presenza di code pesanti e una densità, stimata non parametricamente, che appare diversa da quella della normale.

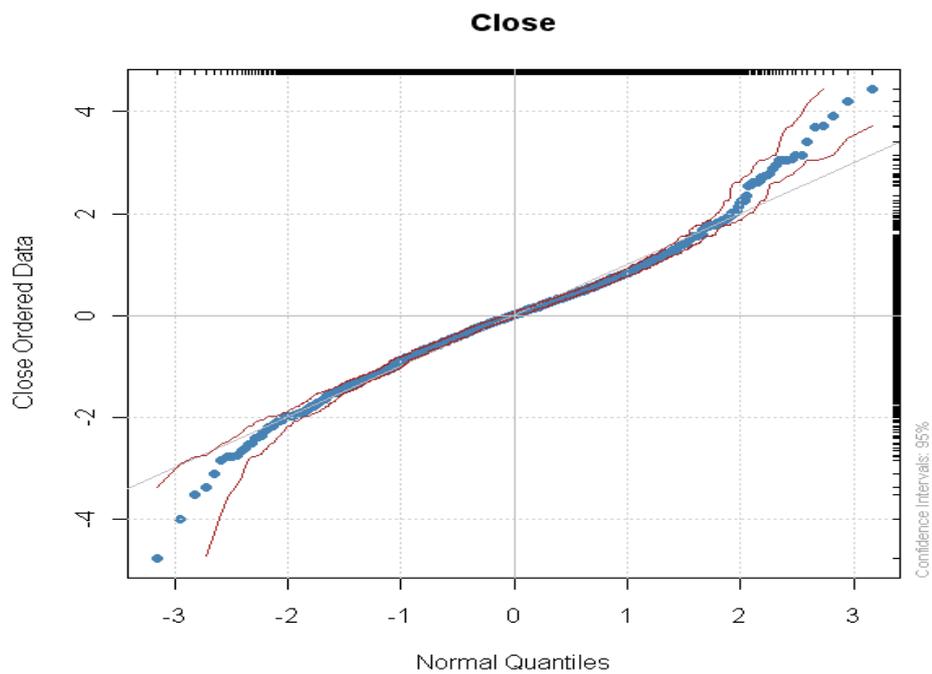


Figure 1.3.1: confronto della distribuzione dei rendimenti con la normale

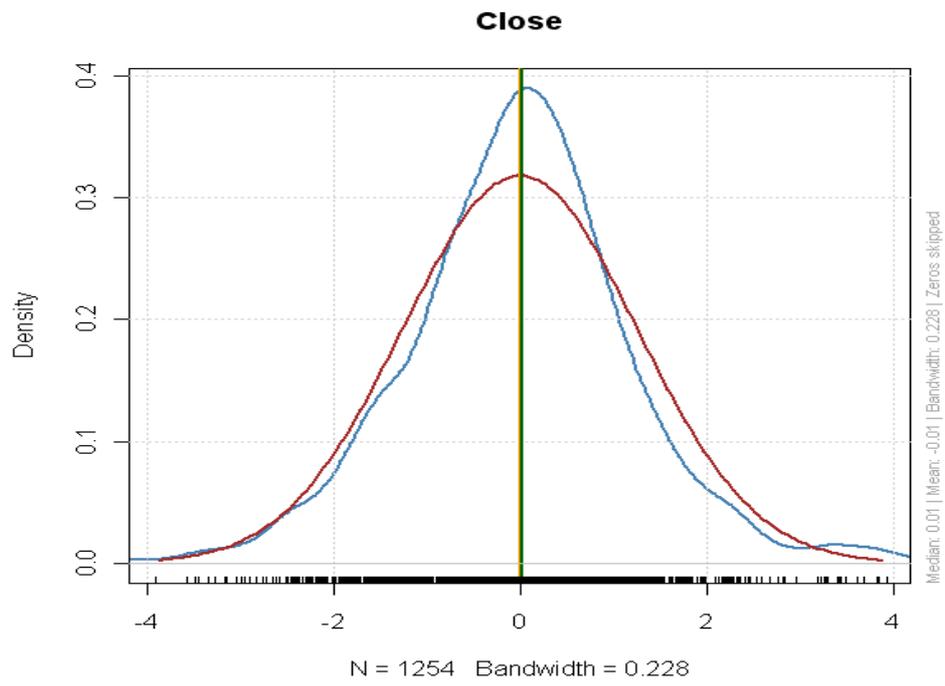


Figure 1.3.2: confronto densità dei rendimenti (in blu) con normale (in rosso)

1.4 Analisi della dipendenza

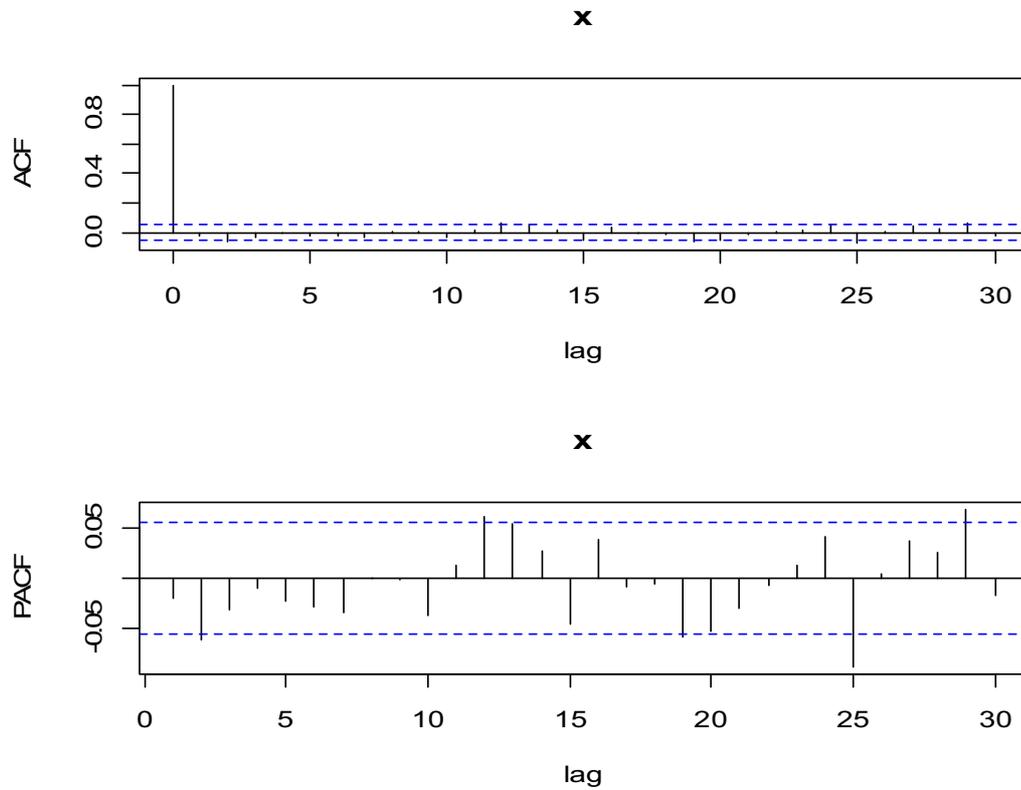


Figure 1.4.1: correlogrammi stimati per la serie dei rendimenti

L'analisi delle funzioni di autocorrelazione totale e parziale della figura ci mostrano una sostanziale assenza di autocorrelazione ai primi ritardi. Esaminando quadrati e valori assoluti dei rendimenti, si nota invece una forte dipendenza:

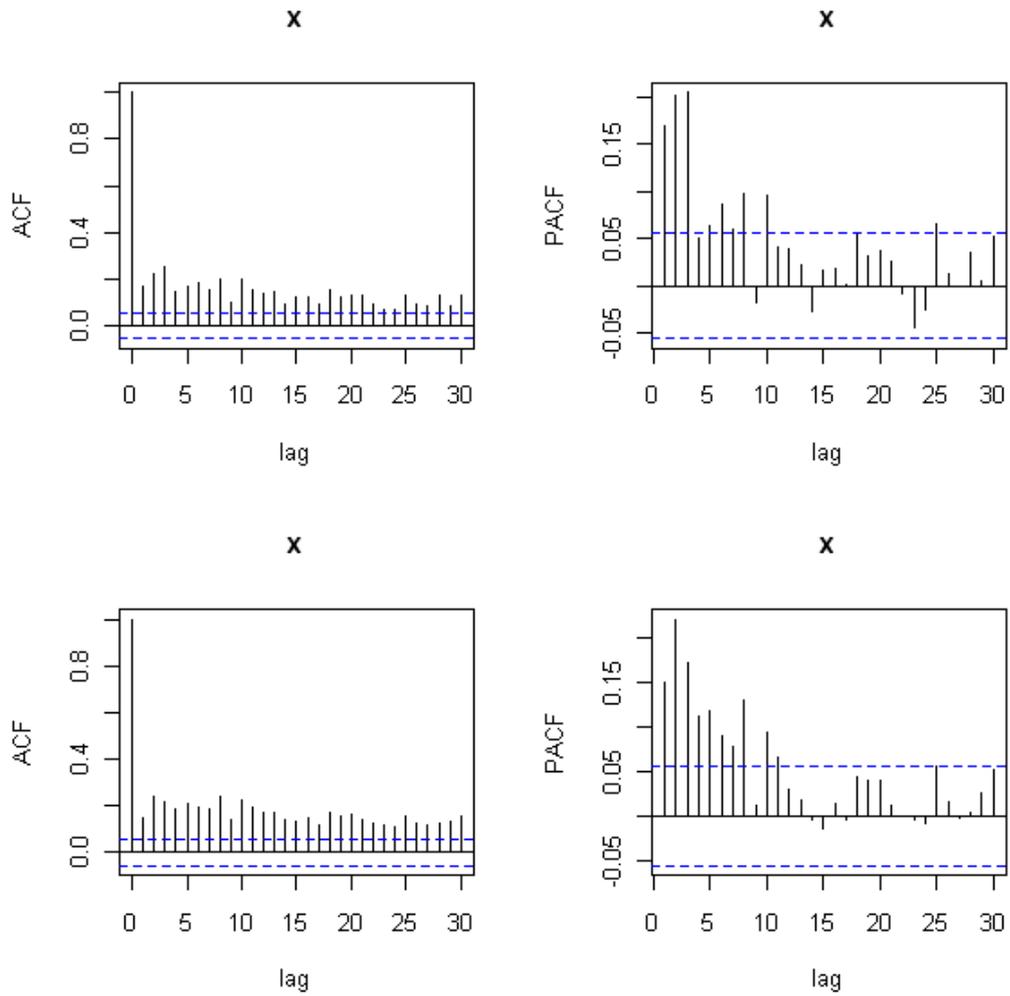


Figure 1.4.2: correlogrammi del quadrato dei rendimenti (sopra) e del valore assoluto (sotto)

Come noto quindi i rendimenti sono generalmente incorrelati, o lo sono debolmente, ma non indipendenti. Ciò ci suggerisce quindi che un modello omoschedastico non va bene.

1.5 Volatilità

Sulla base dei risultati precedenti andremo ad analizzare la volatilità attraverso due modelli utilizzati anche nello studio, GARCH e APARCH.

1.5.1 GARCH(1,1)

Error Analysis:

	Estimate	Std. Error	t value	Pr(> t)
mu	0.025517	0.027346	0.933	0.3508
omega	0.012134	0.006378	1.902	0.0571 .
alpha1	0.079041	0.015525	5.091	3.56e-07 ***
beta1	0.914076	0.016428	55.642	< 2e-16 ***

Questo modello si adatta bene ai dati come mostrato nella tabella seguente. Esso infatti elimina bene l'effetto ARCH (LM Arch Test) e i test di Ljung-Box sui residui e sui quadrati mostra come i residui sono serialmente incorrelati e indipendenti.

Standardised Residuals Tests:

			Statistic	p-Value
Jarque-Bera Test	R	Chi ²	15.20219	0.0004999027
Shapiro-Wilk Test	R	W	0.9961238	0.002968433
Ljung-Box Test	R	Q(10)	6.868485	0.7377997
Ljung-Box Test	R	Q(15)	15.97015	0.3840381
Ljung-Box Test	R	Q(20)	19.19502	0.5091841
Ljung-Box Test	R ²	Q(10)	8.06973	0.6220259
Ljung-Box Test	R ²	Q(15)	11.2076	0.7377439
Ljung-Box Test	R ²	Q(20)	16.35311	0.6944906
LM Arch Test	R	TR ²	10.12696	0.6048232

L'unico problema che rimane è quello della normalità dei residui. Infatti i tests di Jarque-Bera e Shapiro-Wilk rifiutano la nulla che i residui standardizzati siano normali.

1.5.2 APARCH(1,1)

Proviamo ora un modello che tiene conto anche di un eventuale effetto leva.

Error Analysis:

	Estimate	Std. Error	t value	Pr(> t)	
mu	-0.015235	0.023968	-0.636	0.525008	
omega	0.017834	0.004814	3.705	0.000212	***
alpha1	0.056244	0.008231	6.833	8.32e-12	***
gamma1	1.000000	0.004326	231.168	< 2e-16	***
beta1	0.945778	0.010741	88.056	< 2e-16	***
delta	0.755756	0.266059	2.841	0.004504	**

dalle statistiche seguenti vedremo che questo modello al contrario del GARCH accetta le ipotesi di normalità di incorrelazione e di assenza di ulteriori effetti ARCH nei residui.

Standardised Residuals Tests:

			Statistic	p-Value
Jarque-Bera Test	R	Chi ²	1.181078	0.5540286
Shapiro-Wilk Test	R	W	0.9977068	0.07501416
Ljung-Box Test	R	Q(10)	7.5777	0.6700101

Ljung-Box Test	R	Q(15)	17.68836	0.2794037
Ljung-Box Test	R	Q(20)	20.91778	0.4019864
Ljung-Box Test	R ²	Q(10)	12.14166	0.2756873
Ljung-Box Test	R ²	Q(15)	17.64396	0.2818458
Ljung-Box Test	R ²	Q(20)	25.05474	0.1993450
LM Arch Test	R	TR ²	14.85522	0.2494497

Capitolo 2

Introduzione ai modelli di stima della volatilità

La stima della volatilità futura dei rendimenti di un titolo è generalmente ottenuta attraverso due differenti approcci. Il primo è basato su modelli econometrici (Model Based Forecasts). Il secondo modo per stimare la volatilità futura deriva dai prezzi di un'opzione usando la volatilità implicita (IV).

2.1 MBF

I modelli qui utilizzati della famiglia MBF sono estensioni dei modelli della classe ARCH (AutoRegressive Conditional Heteroschedasticity).

Molti modelli per stimare la volatilità si basano sull'assunto che la volatilità sia costante nel tempo. Spesso nella realtà accade che questo assunto sia falso e che non la si possa considerare come una costante a cui aggiungere un semplice errore di campionamento (sampling noise). Nella realtà la volatilità subisce fluttuazioni molto significative, fenomeni come volatility clustering (si veda il grafico dei rendimenti, capitolo 1) stanno ad indicare come periodi diversi presentino diversi gradi di volatilità. In particolare c'è una concentrazione di periodi a bassa volatilità come quelli ad alta.

Engle (1982) suggerisce come l'andamento della varianza del processo generatore sia di tipo autoregressivo: sulla base delle informazioni al

tempo $t-1$ la varianza al tempo t è esprimibile come funzione dei valori più recenti delle innovazioni al quadrato.

In questo lavoro sono state utilizzate tre estensioni e generalizzazioni del semplice modello di classe ARCH. Questi modelli si possono riassumere nella forma più generale descritta dal modello APARCH(m,s):

$$r_t = \mu_t + w_t$$

$$w_t = \sigma_t \varepsilon_t \quad \text{con } \varepsilon_t \sim IID(0,1)$$

e

$$\sigma_t^\delta = \alpha_0 + \sum_{i=1}^m \alpha_i (|w_{t-i}| - \gamma_i w_{t-j})^\delta + \sum_{j=1}^s \beta_j \sigma_{t-j}^\delta$$

Dove $\alpha_0 > 0$, $\delta \geq 0$, $-1 < \gamma_i < 1$, $\alpha_i \geq 0$, $\beta_i \geq 0$.

Con α_i coefficienti della parte autoregressiva riferiti alla parte ARCH del modello e β_i coefficienti dei ritardi della varianza condizionata riferiti alla parte GARCH. Il parametro di potenza δ induce a trasformazioni Box-Cox sulla varianza condizionata. Mentre il parametro γ_i , indicatore dell'effetto leva, produce diversi risultati in caso di "buone notizie" o di "cattive notizie" (w_{t-1} positivo o negativo) in particolare le buone notizie hanno un impatto α_i , mentre le cattive notizie hanno impatto $\alpha_i + \gamma_i$. Con $\sum_{i=1}^{\max(p,q)} \alpha_i + \beta_i < 1$ perchè non "esploda" la varianza non condizionata.

Come si può facilmente verificare gli altri due modelli utilizzati, GARCH e TSGARCH, come è stato detto, sono casi particolari del modello APARCH, in particolare:

il modello GARCH(m,s) lo si ottiene se $\delta=2$, $\gamma_i=0$;

il modello TSGARCH(m,s) lo si ottiene $\delta=1$ e $\gamma_i=0$.

Bisogna anche tenere in mente il problema delle code pesanti della distribuzione dei rendimenti. Per ovviare a questo problema si è provveduto a stimare la distribuzione che più si avvicinava a quella dei rendimenti. La scelta è stata effettuata stimando le distribuzioni che il pacchetto *fGarch* di *R* forniva, e queste sono: *t di Student*, *GED*, *Normale* e rispettive versioni asimmetriche.

2.1.1 RV

Un approccio alternativo alla stima della volatilità è quello di utilizzare dati ad alta frequenza utilizzando la *Realized Volatility* (volatilità realizzata, RV).

Con la volatilità realizzata si stima la volatilità utilizzando l'extra rendimento quadro dei dati intraday. Supponendo di avere n osservazioni equispaziate in un giorno e supponendo che:

$$r_t^d = \sum_{i=1}^n r_{t,i}$$

sia il rendimento giornaliero del titolo, la quantità

$$\hat{\sigma}_d^2 = \frac{n}{n-1} \sum_{i=1}^n (r_{t,i} - \bar{r}_t)^2$$

è definita come la stima della volatilità giornaliera e assumendo media $\bar{r} = 0$ $\hat{\sigma}_d^2 \approx \sum_{i=1}^n r_{t,i}^2$. La somma dei rendimenti logaritmici intra-giornalieri al quadrato è proprio definita come la volatilità realizzata:

$$RV = \sum_{i=1}^n r_{t,i}^2$$

La figura 6 riporta il grafico della RV nel periodo preso in considerazione.

In questo lavoro abbiamo previsto la volatilità attraverso la RV giornaliera del titolo con modelli di serie solitamente utilizzati per la stima della media: il modello ARMA e il modello ARFIMA.

Il modello più generale ARFIMA è espresso dalla seguente formula:

$$AR(B)(1 - B)^d(x_t - \mu_{x_t}) = MA(B)\varepsilon_t$$

Dove AR(B) e MA(B) sono polinomi di ordine rispettivamente p e q . d è il parametro di integrazione frazionaria ($-0.5 < d < 0.5$). Il modello ARMA ha d pari a 0 (per convenzione $(1 - B)^0 = 1$).

Nel contesto è stata utilizzata una trasformazione della serie RV, ossia è stato calcolato $x_t = \ln(\sqrt{RV_t})$. Questa trasformazione è stata attuata in modo da ridurre curtosi e asimmetria nella serie della volatilità realizzata.

Realized Volatility

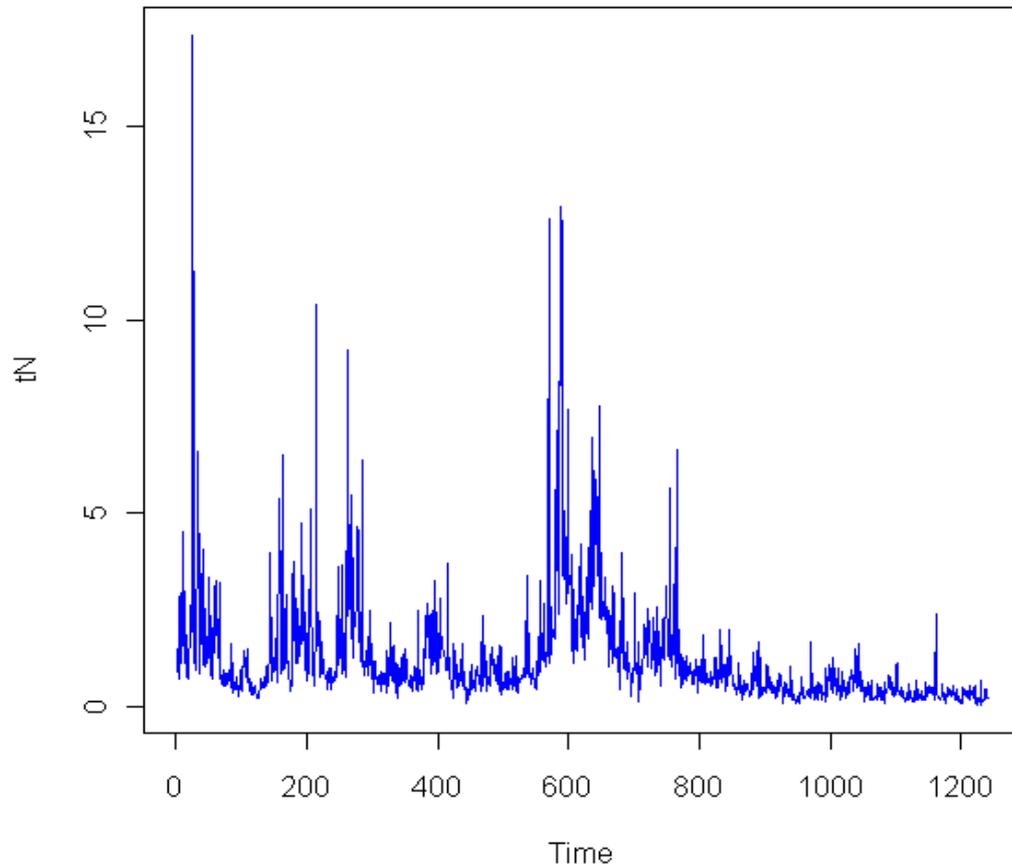


Figure 2.1.1: serie della volatilità realizzata dell'indice S&P 500

	RV_t	x_t
Kurtosis	28.454712	0.223621
Skewness	4.232621	0.088749

Table 2.2.1: curtosi e asimmetria per la volatilità realizzata e la sua trasformazione

Per la stima della volatilità attraverso la RV, e in particolare la sua trasformazione x_t , sono stati utilizzati i modelli ARMA(2,1) e ARFIMA(1,d,0).

Il modello ARFIMA(p,d,q) presenta però problemi per quanto concerne le previsioni. Nello specifico non c'è nessuna funzione in R che permette di calcolare le previsioni derivanti da un modello ARFIMA. Alcuni studi, *Corduas (1999, 2000)*, hanno dimostrato come un modello ARFIMA(p,d,q) con $d > 0$, possa essere approssimato mediante un modello ARMA(p+1, q+1) al fine di poter utilizzare le previsioni di quest'ultimo; ciò avviene sostituendo all'operatore differenza frazionaria $(1 - B)^d$ l'operatore $(1 - \Phi B)(1 - \Theta B)^{-1}$ tale approssimazione rende i due modelli il più possibile "vicini", *D'Elia-Piccolo (2002)*. Nel nostro caso utilizzando l'approssimazione, si ottiene un modello ARMA(2,1) che è proprio il modello che già si utilizzava.

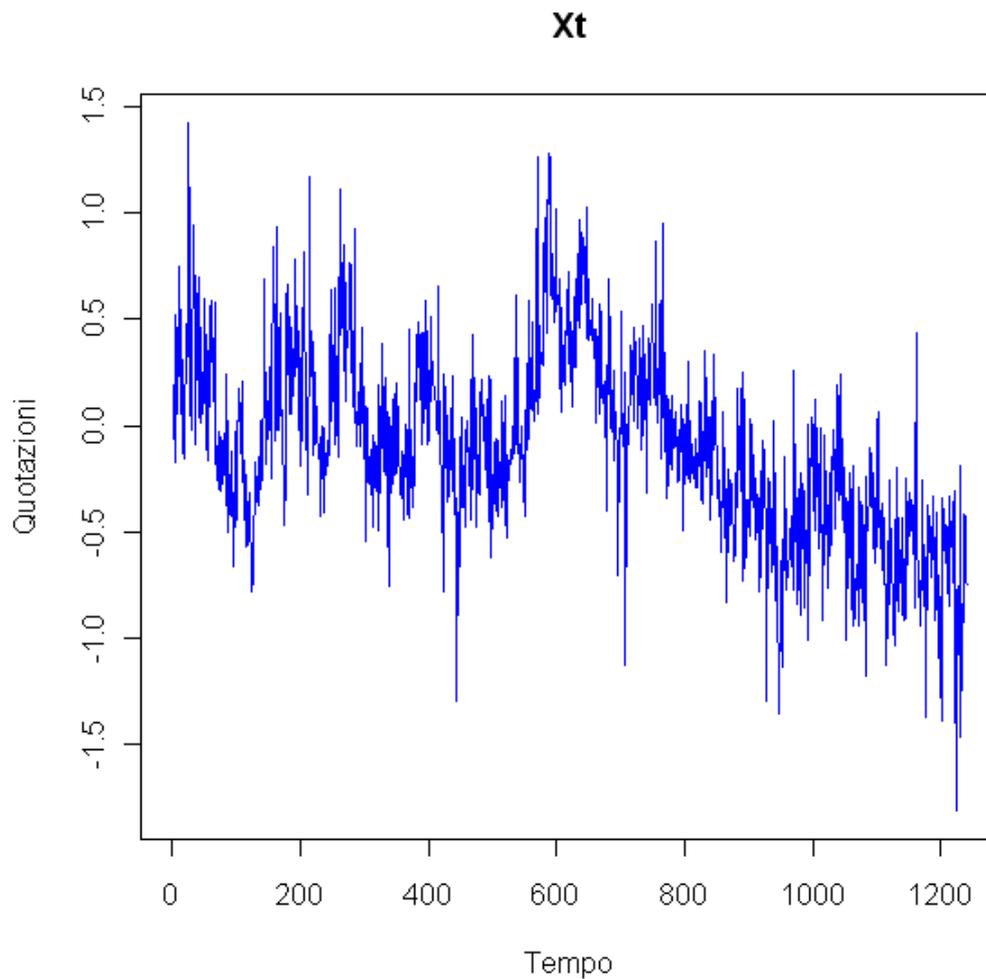


Figure 2.1.2: serie della trasformazione utilizzata per stimare la volatilità realizzata

2.2 IV-VIX

Nel determinare il prezzo di un'opzione, il grado di incertezza dell'attività sottostante è un elemento determinante tanto che dalle quotazioni delle opzioni a vari prezzi di esercizio si possono ricavare le volatilità implicite (IV) che corrispondono alla migliore previsione della volatilità di un'asset fornita da parte dei mercati.

Il VIX (Volatility Index) costruito dalla CBOE (Chicago Board Options Exchange) costituisce la stima di volatilità impiegata sulle opzioni dello S&P 500 utilizzata in questo lavoro.

Mentre il vero processo sottostante il prezzo di un'opzione è sconosciuto il VIX è costruito per essere una misura generale della volatilità media stimata in una sottosequenza di 22 giorni lavorativi.

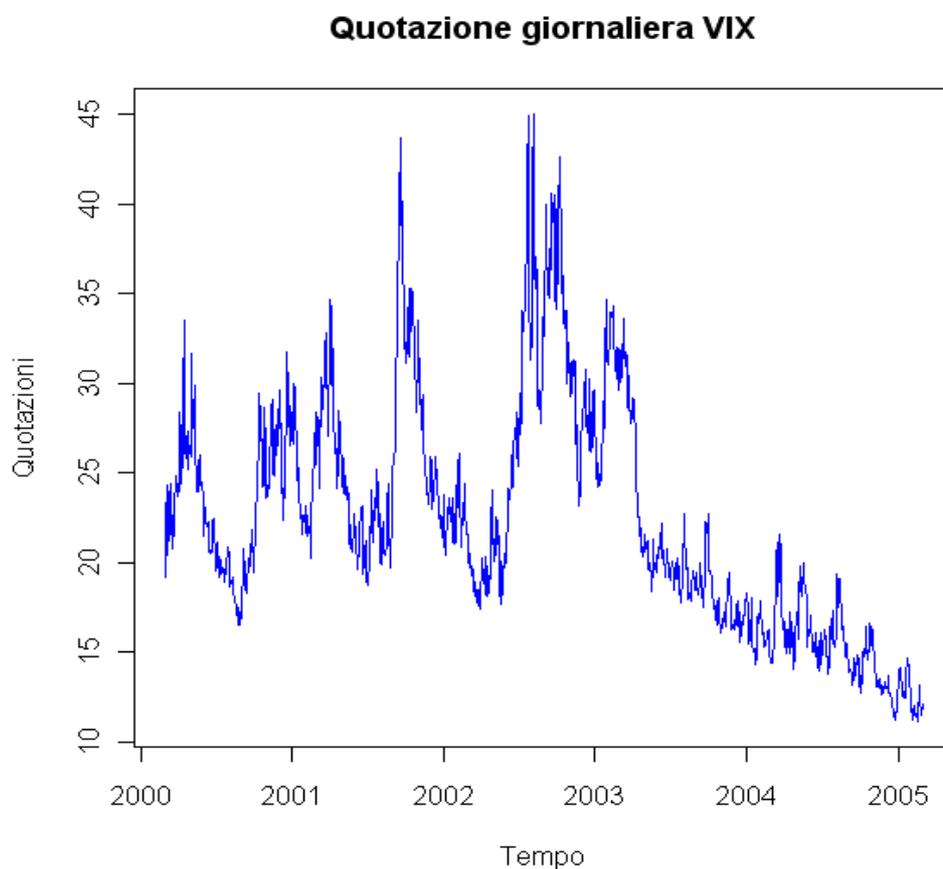


Figure 2.2.1: serie giornaliera del VIX

Capitolo 3

Modelli e combinazioni

Prima di poter valutare se è meglio prevedere la volatilità con una previsione individuale o con una combinazione di previsioni, in questo capitolo si descriverà come si è arrivati alle stime della volatilità futura utilizzate poi per le combinazioni.

Per generare una sequenza di previsioni della volatilità basata sull'informazione al tempo t , i modelli per la volatilità sono stati ristimati per diversi passi utilizzando dati dal tempo $t-749$ al tempo t . I modelli così ottenuti sono stati utilizzati poi per la previsione della volatilità a 22 giorni (dal tempo $t+1$ al tempo $t+22$). Questo periodo corrisponde al periodo coperto dal VIX, che come detto nel capitolo 2 è la stima di volatilità dell'indice S&P 500 che il mercato offre su un periodo di 22 giorni. Il primo periodo di previsione va dal 18 marzo 2003 al 16 aprile 2003. Per ogni sottosequenza i parametri dei modelli sono stati ristimati utilizzando una *rolling estimation window* di 750 osservazioni. L'ultimo periodo di previsione va dal 27 gennaio 2005 al 28 febbraio 2005 per un totale di 470 previsioni per ogni modello trattato.

3.1 Come combinare le previsioni

E' possibile utilizzare due modi per costruire una combinazione di previsioni. Il più semplice è quello di combinare le previsioni tutte con lo stesso peso; non sarà però questo il modo che utilizzeremo per fare una combinazione.

Il secondo modo, introdotto da *Clements e Hendry (1998)*, è quello di formare una combinazione lineare pesata con pesi forniti dalla seguente regressione:

$$\overline{RV}_{t+22} = \alpha_0 + \alpha_1 f_t^1 + \alpha_2 f_t^2 + \dots + \alpha_n f_t^n + \varepsilon_t,$$

dove \overline{RV}_{t+22} è la volatilità target, cioè la media della volatilità realizzata nell'orizzonte previsivo di 22 giorni (da $t+1$ a $t+22$), ed f_t^i con $i=1, 2, \dots, n$ sono le medie delle previsioni di volatilità nei 22 giorni stimati fino al tempo t (da $t+1$ a $t+22$). La risultante combinazione sarà data da

$$f_t^C = \widehat{\alpha}_0 + \widehat{\alpha}_1 f_t^1 + \widehat{\alpha}_2 f_t^2 + \dots + \widehat{\alpha}_n f_t^n$$

dove f_t^C indica la combinazione delle previsioni, mentre $\widehat{\alpha}_i$ $i=0,1,\dots,n$ sono le stime OLS dei parametri del modello di regressione. I modelli utilizzati per le previsioni sono quelli descritti nel capitolo 2 e cioè: ARMA, stimato sulla volatilità realizzata, GARCH, APARCH e TSGARCH, stimati sui rendimenti, nella combinazione di previsioni viene utilizzata anche la serie del VIX. Le combinazioni utilizzate saranno presentate nel capitolo 4.

APPENDICE 1

Nell'appendice che segue è stata riportata la parte di programma che si occupa di fare ciò che è stato descritto nel capitolo. In particolare ciò che si andrà a fare è dividere la serie di rendimenti in piccole sottosequenze (le finestre) e su ogni sottosequenza stimare la volatilità attraverso i modelli che ci siamo prefissati di utilizzare.

Per ogni finestra costituita da 750 rendimenti, per prima cosa è stata stimata la sua distribuzione tra quelle che il pacchetto fGarch di R fornisce (ossia Normale, t di Student, GED e rispettive versioni asimmetriche). Tra queste la distribuzione è stata scelta utilizzando il criterio dell'AIC attraverso la formula: $AIC = 2 * \ln(l) + 2 * k$, dove l è il valore massimizzato della funzione di verosimiglianza e k è il numero di parametri. Questa distribuzione sarà poi utilizzata come distribuzione per i modelli GARCH APARCH e TSGARCH per una migliore previsione.

Dopo la scelta della distribuzione si è passati alla stima vera e propria delle previsioni di volatilità. Come detto nel capitolo 2, utilizzeremo il modello ARMA per la previsione della volatilità tramite i dati della volatilità realizzata del titolo e in particolare come dato utilizzeremo la sua trasformazione x_t ; per i restanti modelli della classe APARCH i dati utilizzati sono i rendimenti logaritmici. Per ogni finestra sono stati presi 750 valori, è stato stimato il modello opportuno e sui valori previsti è stata calcolata la media delle 22 previsioni di volatilità dal tempo $t+1$ al tempo $t+22$.

Per ottenere la media delle 22 previsioni del VIX, che deve ovviamente corrispondere allo stesso periodo degli altri modelli, è stata calcolata la

media dall'osservazione 751-esima all'osservazione 772-esima di ogni finestra, essendo esso già una misura di volatilità. Stesso discorso e stesso intervallo di valori per il calcolo della media della volatilità realizzata che sarà poi la variabile target.

Finito questo si sposta in avanti la finestra di un'osservazione e si ripete tutto per 470 volte tramite un ciclo *for*.

Tutte le 470 medie ottenute da ogni modello sono state salvate in un array col nome del modello che le ha generate e saranno poi utilizzate come variabili per la regressione descritta nel capitolo.

Estratto del codice R che si occupa delle previsioni di volatilità

```
for(i in 1:470)
{
  g=i+749
  a=750+i
  b=771+i

  ##STIMA DELLA DISTRIBUZIONE DELLA SOTTOSEQUENZA DEI
  RENDIMENTI

  fit.norm = normFit(r1[i:g]) # Normale
  fit.std = stdFit(r1[i:g]) # t di Student
  fit.ged = gedFit(r1[i:g]) # GED
  fit.sstd = sstdFit(r1[i:g]) # t di Student
  asimmetrica
  fit.sged = sgedFit(r1[i:g]) # GED asimmetrica

  fit.hyp = hypFit(r1[i:g],doplot=F,trace=F)
  #Iperbolica

  fit.nig = nigFit(r1[i:g],doplot=F,trace=F)
  #Gaussiana Inversa

  fit.snorm = snormFit(r1[i:g]) #Normale asimmetrica
```

```

aic.norm = 2*fit.norm$minimum + 2*2/T
aic.std = 2*fit.std$minimum + 2*3/T
aic.ged = 2*fit.ged$minimum + 2*3/T
aic.sstd = 2*fit.sstd$minimum + 2*4/T
aic.sged = 2*fit.sged$minimum + 2*4/T
aic.hyp = 2*abs(fit.hyp@fit$minimum) + 2*4/T
aic.nig = 2*abs(fit.nig@fit$minimum) + 2*4/T
aic.snorm = 2*fit.snorm$minimum + 2*3/T

```

```

#####SCELTA DELLA MIGLIORE
DISTRIBUZIONE#####

```

```

min.aic<-
min(c(aic.norm,aic.std,aic.ged,aic.sstd,aic.sged,
aic.snorm))

```

```

if(min.aic-aic.norm==0) distarr=dis="dnorm"
if(min.aic-aic.std==0) distarr=dis="dstd"
if(min.aic-aic.ged==0) distarr=dis="dged"
if(min.aic-aic.sstd==0) distarr=dis="dsstd"
if(min.aic-aic.sged==0) distarr=dis="dsged"
if(min.aic-aic.snorm==0) distarr=dis="dsnorn"

```

```

#####PREVISIONI#####

```

```

arma[i]=mean(predict(armaFit(~arma(2,1),
data=xt[i:g]),22)$pred)

aparch[i]=mean(predict(garchFit(~aparch(1,1),
data=r1[i:g], cond.dist=dis),22)$standardDeviation)

garch[i]=mean(predict(garchFit(~garch(1,1),
data=r1[i:g], cond.dist=dis),22)$standardDeviation)

tsgarch[i]=mean(predict(garchFit(~garch(1,1),
data=r1[i:g], cond.dist=dis,
delta=1),22)$standardDeviation)

vvix[i]=mean(vix1[a:b])

rvmean[i]=mean(tN[a:b])
}

```


Capitolo 4

Valutazione delle previsioni: MSE, QLIKE e Diebold-Mariano Test

Una volta ottenute le previsioni di volatilità, per determinare quale di queste previsioni o quale combinazione di previsioni risulta migliore ci serviremo di due strumenti: uno è basato su funzioni di perdita, metodo introdotto da Hansen (2003); l'altro è l'utilizzo del test di Diebold e Mariano.

Le due funzioni di perdita utilizzate sono MSE e QLIKE:

$$MSE^i = \frac{1}{T} \sum_{t=1}^T (\overline{RV}_{t+22} - f_t^i)^2$$

$$QLIKE^i = \frac{1}{T} \sum_{t=1}^T \left[\log(f_t^i) + \frac{\overline{RV}_{t+22}}{f_t^i} \right]$$

dove f_t^i sono le previsioni al tempo t ottenute dai modelli presi individualmente, dal VIX e dalle combinazioni di due o più modelli con pesi stimati dalla regressione.

MSE e QLIKE danno però solo un'informazione descrittiva su quale sia il modello migliore ma non ci permettono di verificare se previsioni ottenute due diversi modelli hanno differenze statisticamente significative; utilizzeremo, per questo, il test di Diebold e Mariano. Nello specifico l'ipotesi nulla che testa è la seguente:

$$E[L(e_t^i) - L(e_t^j)] = 0$$

Dove $L()$ indica la funzione di perdita utilizzata ed $e_t^i = (\overline{RV}_{t+22} - f_t^i)$ è l'errore di previsione i -esimo.

4.1 Singole previsioni

	MSE		QLIKE
ARMA	0.02312718	ARMA	0.3302311
GARCH	0.02354186	TSGARCH	0.3330201
TSGARCH	0.02374258	GARCH	0.3384206
APARCH	0.03223364	APARCH	0.3415538
VIX	0.07440224	VIX	0.3926462

Table 4.1.1: MSE e QLIKE delle previsioni individuali

La tabella 4.1.1 mostra MSE e QLIKE relativi alle previsioni individuali. La previsione della volatilità ottenuta con il modello ARMA stimato sulla Volatilità Realizzata giornaliera produce una perdita media inferiore rispetto alle altre previsioni sia utilizzando il criterio MSE che con il criterio QLIKE, produce cioè una migliore previsione di \overline{RV}_{t+22} . A seguire ci sono le previsioni ottenute attraverso i modelli GARCH e TSGARCH, mentre le previsioni basate sul VIX sono le peggiori.

Utilizzando ora il test di Diebold-Mariano, ad una coda con orizzonte temporale pari ad un mese, verifichiamo se le differenze tra le previsioni ottenute dai vari modelli sono statisticamente significative. Testeremo in particolare l'ipotesi nulla che:

$$E[L(e_t^i) - L(e_t^j)] \geq 0 \text{ con } i \neq j$$

contro l'alternativa

$$E[L(e_t^i) - L(e_t^j)] < 0$$

Dalla tabella 4.1.2 possiamo notare come il VIX che occupava l'ultima posizione nella tabella 4.1.1 è il modello che effettivamente produce una previsione di volatilità mediamente peggiore rispetto a tutte le altre con un *p-value* al di sotto del convenzionale livello di significatività del 5%. Stesso discorso si può fare per le previsioni ottenute dal modello APARCH che produce stime migliori solo rispetto al VIX.

La tabella riporta sia i *p-value* ottenuti dal confronto tra le previsioni *i-esime* e le previsioni *j-esime*, sia il confronto tra le *j-esime* con le *i-esime*. Questo è stato fatto per rendere la tabella più immediata anche se è una ridondanza. In particolare è stato fatto per far venire all'occhio un fatto in particolare, per esempio: il *p-value* del confronto tra l'APARCH con il GARCH è del 6% che porta a dire che la nulla non è rifiutata a livello di significatività del 5%; il test del confronto tra GARCH e APARCH fornisce un *p-value* del 94%, ancora, la nulla è non rifiutata. La conclusione che ne deriva è che: i modelli producono previsioni statisticamente non diverse però c'è una *tendenza* a preferire le previsioni ottenute dal modello GARCH, perché il *p-value* è più alto.

Ancora facendo collegamenti con la tabella 4.1.1 si nota che le previsioni ARMA siano quelle che forniscono effettivamente un *p-value* superiore rispetto a tutte le altre. Notiamo che c'è una differenza statistica tra le previsioni ARMA e quelle ottenute dal VIX e dall'APARCH, *p-value* superiore al 5%, e che c'è una preferenza tendenziale contro le previsioni GARCH e TSGARCH.

i \ j	GARCH	TSGARCH	ARMA	APARCH	VIX
GARCH	–	54,1	44.4	94,0	97,6
TSGARCH	45,9	–	41.6	97,5	98,3
ARMA	55.6	58,4	–	97,4	97,9

APARCH	6,0	2,5	2,6	-	97,4
VIX	2,4	1,7	2,1	2,6	-

Table 4.1.2: *p-value* del test di Diebold-Mariano: il modello in riga è migliore di quello in colonna.
%

4.2 Previsioni combinate

Qui andremo a combinare un pò di previsioni per verificare se c'è una combinazione che ci fornisce un risultato più accurato, rispetto alle singole previsioni.

Alcune delle combinazioni che sono state trattate sono: ARMA+VIX, APARCH+GARCH+TSGARCH, tutti modelli della famiglia APARCH, e la combinazione che comprende tutt'e quattro le previsioni ottenute dagli altrettanti modelli trattati. Di queste ultime due combinazioni sono state considerate anche le versioni con l'aggiunta del VIX, in particolare l'ultimo che comprende tutte le previsioni dei modelli trattati è una sorta di modello completo. Come ultima combinazione è stata considerata ARMA+VIX+TSGARCH, combinazione ottenuta togliendo dal modello completo le variabili con peso stimato non significativamente diverso da zero¹.

Sia l' MSE che il QLIKE mostrano come la combinazione di tutte e cinque le previsioni di volatilità procuri una perdita media minore rispetto alle altre combinazioni e rispetto a tutte le previsioni ottenute da un singolo modello. La combinazione ARMA+VIX+GARCH occupa la posizione

¹Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	-0.15075	0.05711	-2.639	0.00859	**						
I(exp(2 * arma))	0.46864	0.06298	7.441	4.89e-13	***						
garch	0.14350	0.15465	0.928	0.35396							
tsgarch	0.23573	0.11392	2.069	0.03909	*						
aparch	0.03421	0.08388	0.408	0.68354							
vvix	0.05754	0.01183	4.864	1.58e-06	***						

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

successiva con valori che si discostano di poco dal modello completo, addirittura per il QLIKE è anche migliore. A seguire troviamo ancora le combinazioni delle previsioni di tutti i modelli senza VIX e la combinazione tra ARMA e VIX a posizioni invertite nei due criteri. Sempre più giù poi troviamo le previsioni ottenute dai singoli modelli e a chiudere quelle ottenute col VIX. Si può notare da questa tabella che tutte le combinazioni trattate producano una perdita media inferiore rispetto alle previsioni ottenute dai modelli singoli.

	MSE		QLIKE
ALL+VIX	0.01952861	ARMA+VIX+ TSGARCH	0.3255519
ARMA+VIX+ TSGARCH	0.01957009	ALL+VIX	0.3260079
ALL	0.02052413	ARMA+VIX	0.3285091
ARMA+VIX	0.02153440	ALL	0.3291429
APARCH+GARCH+ TSGARCH+VIX	0.02185889	APARCH+GARCH+ TSGARCH+VIX	0.3293523
APARCH+GARCH+ TSGARCH	0.02230531	ARMA	0.3302311
ARMA	0.02312718	APARCH+GARCH+ TSGARCH	0.3318505
GARCH	0.02354186	TSGARCH	0.3330201
TSGARCH	0.02374258	GARCH	0.3384206
APARCH	0.03223364	APARCH	0.3415538
VIX	0.07440224	VIX	0.3926462

Table 4.2.1: MSE e QLIKE delle previsioni individuali e combinate

L'analisi del test di Diebold-Mariano ci mostra ancora una volta, e a conferma dei criteri MSE e QLIKE, che il VIX fornisce le peggiori previsioni rispetto a tutti i modelli trattati. Inoltre, le tre combinazioni che occupano le prime posizioni sono realmente quelle che forniscono i risultati statisticamente più significativi sia nel confronto con le previsioni ottenute dai singoli modelli sia rispetto alle previsioni ottenute dalle altre

combinazioni. Ed effettivamente la combinazione ottenuta da tutti e cinque i modelli offre, tendenzialmente, la previsione migliore anche rispetto alle previsioni ottenute da ARMA+VIX e da quelle ottenute da tutti i modelli senza VIX. Buone previsioni sono fornite anche dal modello ARMA+VIX+TSGARCH.

$i \setminus j$	GARCH	TSGARCH	ARMA	APARCH	VIX	ARMA+VIX	GARCH+TSGARCH+APARCH	ALL	APARCH+GARCH+TSGARCH+VIX	ALL+VIX	ARMA+VIX+TSGARCH
GARCH	-	54,1	44,4	94,0	97,6	25,4	13,4	4,0	17,8	3,1	3,8
TSGARCH	45,9	-	41,6	97,5	98,3	18,8	19,4	9,4	9,6	2,4	2,1
ARMA	55,6	58,4	-	97,4	97,9	16,4	36,3	5,1	31,8	3,1	2,9
APARCH	6,0	2,5	2,6	-	97,4	0,6	2,2	1,4	1,1	0,4	0,4
VIX	2,4	1,7	2,1	2,6	-	1,5	2,0	1,4	1,7	1,9	1,4
ARMA+VIX	74,7	81,2	83,6	99,4	98,5	-	63,5	31,7	56,8	5,2	3,7
GARCH+TSGARCH+APARCH	86,6	80,6	63,7	97,8	98,0	36,5	-	7,9	33,0	1,8	2,6
ALL	96,0	90,6	94,9	98,6	98,1	68,3	92,1	-	78,9	22,8	25,2
APARCH+GARCH+TSGARCH+VIX	82,2	90,4	68,2	98,9	98,3	43,2	67,0	25,1	-	2,6	2,9
ALL+VIX	97,0	97,7	96,9	99,6	98,6	94,8	98,3	77,2	97,4	-	58,1
ARMA+VIX+TSGARCH	96,2	97,9	97,1	99,6	98,6	96,3	97,4	74,8	97,1	41,9	-

Table 4.2.2: *p-value* del test di Diebold-Mariano: il modello in riga è migliore del modello in colonna. %

Oltre ad utilizzare i risultati del test di Diebold-Mariano facciamo un confronto anche con quelli della tabella dall'AIC (Akaike Information Criterion) per individuare il modello, che oltre a fornire le previsioni migliori, è il più parsimonioso.

Notiamo che le previsioni ottenute dalla combinazione di ARMA, VIX e TSGARCH oltre ad essere delle migliori sono anche ottenute da una regressione lineare più parsimoniosa, essa infatti occupa la prima posizione nella tabella dell'AIC.

Ovviamente la combinazione ARMA+VIX+TSGARCH è migliore rispetto al modello completo per costruzione, esso infatti è privo delle previsioni non significative.

I risultati comunque vanno in sintonia con quanto fin qui descritto.

	AIC
ARMA+VIX+TSGARCH	-505.0617
ALL+VIX	-502.059
ALL	-480.6901
APARCH+GARCH+TSGARCH+VIX	-451.0771
APARCH+GARCH+TSGARCH	-443.5752
AIC ARMA	-430.5688
GARCH	-422.2161
TSGARCH	-418.2258
APARCH	-274.5277
ARMA+VIX	-272.0627
VIX	118.6157

Table 4.2.3: tabella dell'AIC, partendo dall'alto sono riportati i modelli migliori

Conclusioni

In questo studio ci si è chiesti qual è il modo migliore per prevedere la volatilità. Abbiamo utilizzato previsioni ottenute da quattro diversi modelli econometrici, presi individualmente; previsioni derivanti da uno strumento, il VIX, costruito dal mercato, e sei previsioni ottenute dalla combinazione di questi modelli. In tutto sono stati passati in esame undici modi diversi di prevedere la volatilità.

La prima cosa che ci salta all'occhio da questo studio effettuato sullo S&P 500, è che l'utilizzo del VIX, misura della volatilità implicita dello S&P 500, è il peggiore modo di produrre previsioni. Esso infatti, risulta lo strumento che procura previsioni statisticamente inferiori rispetto agli altri modelli, la conferma statistica ce la dà test di *Diebold-Mariano*. È il modello che fornisce il *p-value* peggiore confrontandolo con tutte le altre previsioni siano esse derivanti da singole previsioni o dalla loro combinazione.

La previsione migliore proveniente da un singolo modello ce la fornisce l'ARMA che utilizza la Volatilità Realizzata giornaliera dell'indice S&P 500. Esso infatti è quello che provoca una minore e significativa perdita media rispetto alle altre previsioni.

Quando andiamo però a combinare i modelli con pesi stimati da una regressione ci si accorge che questa produce risultati significativamente migliori rispetto alle previsioni provenienti dai modelli singoli, ogni

combinazione produce un'accuratezza superiore rispetto alle previsioni che la compongono.

La previsione migliore, di quelle utilizzate, è ottenuta dalla combinazione di previsioni ottenute da VIX, ARMA e dal TSGARCH. Tre previsioni ottenute da tre tipi di dati, volatilità implicita dello S&P 500 il primo, Volatilità realizzata giornalmente dal titolo il secondo e volatilità ottenuta dalla serie storica dei rendimenti il terzo. La presenza del VIX in questa combinazione, quindi, non deve stupire, le tre previsioni ottenute dai tre modelli producono tre diverse differenti dinamiche di volatilità.

Nella sostanza le conclusioni ottenute in questo lavoro non si discostano di molto da quelle ottenute in Becker e Clements (2008), pur utilizzando un diverso insieme di dati e con riferimento ad un periodo temporale differente.

Codice programma

Di seguito è riportato il codice utilizzato per lo studio. In particolare la prima parte si riferisce all'analisi del capitolo 1, mentre la seconda parte si riferisce allo studio vero e proprio.

1. Codice per analisi del capitolo 1

```
library(tseries)
library(Rmetrics)
library(fracdiff)
library(FinTS)
tsc=get.hist.quote(instrument="^GSPC",
                  start="2000-02-29", end="2005-02-28",
                  quote=c("Close"), compression=c("d"),
                  retclass="zoo")
vix=get.hist.quote(instrument="^VIX",
                  start="2000-02-29", end="2005-02-28",
                  quote=c("AdjClose"), compression=c("d"),
                  retclass="zoo")

rv=read.table("RV2.txt", header=TRUE)
N=rv$RV
tN=as.ts(N)
class(tN)
xt=log(sqrt(tN)) ###logaritmo della radice di rv
str(tN)
length(ts)
```

```

plot(tsc, type="l", main="Quotazione giornaliera S&P500",
col=4,
      xlab="Tempo", ylab="Quotazioni")

#####RENDIMENTI#####

r=diff(log(tsc))*100

r1=c(r[1:263],r[265:386],r[398:719],r[721:982],r[984:length(r
)])

vix1=c(vix[1:263],vix[265:386],vix[398:719],vix[721:982],vix[
984:length(vix)])

plot(r1, type="l", main="Rendimenti logaritmici S&P500",
col=4,
      xlab="Tempo", ylab="Quotazioni")

plot(rvix, type="l", main="Rendimenti logaritmici VIX",
col=4,
      xlab="Tempo", ylab="Quotazioni")

basicStats(r)

basicStats(rvix)

r=coredata(r)

acfPlot(r)

pacfPlot(r)

par(op)

#####STIMA
DENSITA'##### DELLA

round(basicStats(r),digit=3)

```

```

normalTest(r, method = "sw")
normalTest(r, method = "ks")
normalTest(r, method = "da")

op=par(mfrow=c(3,1))
qqnormPlot(r)
histPlot(r)
densityPlot(r)
par(op)

## analisi della varianza
fit1=garchFit(~garch(1,1), data=r, cond.dist="std")
summary(fit1)

fit2=garchFit(~aparch(1,1), data=r, cond.dist="std")
summary(fit2)

```

2. Codice per previsioni, regressioni per combinazioni e test Diebold-Mariano

```

## definizione variabili
arma=0
aparch=0
arfima=0
garch=0
gjr=0
vvix=0
tsgarch=0
rvmean=0
tsgarch=as.array(tsgarch)

```

```

arma=as.array(arma)
aparch=as.array(aparch)
arfima=as.array(arfima)
garch=as.array(garch)
vvix=as.array(vvix)
gjr=as.array(gjr)
rvmean=as.array(rvmean)
dis="dnorm"
distarr=as.array

## ciclo for per la stima delle previsioni di volatilità
for(i in 1:470)
{
  g=i+749
  a=750+i
  b=771+i

## stima della distribuzione della finestra di rendimenti
  fit.norm = normFit(r1[i:g]) # Normale
  fit.std = stdFit(r1[i:g]) # t di Student
  fit.ged = gedFit(r1[i:g]) # GED
  fit.sstd = sstdFit(r1[i:g]) # t di Student
asimmetrica
  fit.sged = sgedFit(r1[i:g]) # GED asimmetrica
  fit.hyp = hypFit(r1[i:g],doplot=F,trace=F)
#Iperbolica
  fit.nig = nigFit(r1[i:g],doplot=F,trace=F) #Gaussiana
Inversa
  fit.snorm = snormFit(r1[i:g]) #Normale asimmetrica

aic.norm = 2*fit.norm$minimum + 2*2/T
  aic.std = 2*fit.std$minimum + 2*3/T

```

```

aic.ged = 2*fit.ged$minimum + 2*3/T
aic.sstd = 2*fit.sstd$minimum + 2*4/T
aic.sged = 2*fit.sged$minimum + 2*4/T
aic.hyp = 2*abs(fit.hyp@fit$minimum) + 2*4/T
aic.nig = 2*abs(fit.nig@fit$minimum) + 2*4/T
aic.snorm = 2*fit.snorm$minimum + 2*3/T

## scelta migliore distribuzione
min.aic<-min(c(aic.norm,aic.std,aic.ged,aic.sstd,aic.sged,
aic.snorm))

if(min.aic-aic.norm==0) distarr=dis="dnorm"
  if(min.aic-aic.std==0) distarr=dis="dstd"
  if(min.aic-aic.ged==0) distarr=dis="dged"
  if(min.aic-aic.sstd==0) distarr=dis="dsstd"
  if(min.aic-aic.sged==0) distarr=dis="dsged"
  if(min.aic-aic.snorm==0) distarr=dis="dsnorm"

## stima della volatilità attraverso i vari modelli

arma[i]=mean(predict(armaFit(~arma(2,1),
data=xt[i:g]),22)$pred)####

aparch[i]=mean(predict(garchFit(~aparch(1,1), data=r1[i:g],
cond.dist=dis),22)$standardDeviation)

garch[i]=mean(predict(garchFit(~garch(1,1), data=r1[i:g],
cond.dist=dis),22)$standardDeviation)##ok

tsgarch[i]=mean(predict(garchFit(~garch(1,1), data=r1[i:g],
cond.dist=dis, delta=1),22)$standardDeviation)####ok

```

```

vvix[i]=mean(vix1[a:b])

rvmean[i]=mean(tN[a:b])
}

#####FIT MSE E QLIKE#####

## generazione delle combinazioni di previsioni
fitArma=lm(rvmean~I(exp(2*arma)))
fitGarch=lm(rvmean~garch)
fitTsgarch=lm(rvmean~tsgarch)
fitAparch=lm(rvmean~aparch)
fitVvix=lm(rvmean~vvix)
fitAllVix=lm(rvmean~I(exp(2*arma))+garch+tsgarch+aparch+vvix)
fitAll=lm(rvmean~I(exp(2*arma))+garch+tsgarch+aparch)
fitApfam=lm(rvmean~garch+aparch+tsgarch)
fitArVvix=lm(rvmean~I(exp(2*arma))+vvix)
fitApfamVvix=lm(rvmean~garch+aparch+tsgarch+vvix)

## calcolo MSE e QLIKE per ogni combinazione
mseArma=mean((rvmean-predict(fitArma))^2)
qlikeArma=mean(log(predict((fitArma))+rvmean/(predict(fitArma))))

mseGarch=mean((rvmean-predict(fitGarch))^2)
qlikeGarch=mean(log(predict(fitGarch))+rvmean/predict(fitGarch))

mseTsgarch=mean((rvmean-predict(fitTsgarch))^2)
qlikeTsgarch=mean(log(predict(fitTsgarch))+rvmean/predict(fitTsgarch))

```

```

mseVvix=mean((rvmean-predict(fitVvix))^2)
qlikeVvix=mean(log(predict(fitVvix))+rvmean/predict(fitVvix))

mseAparch=mean((rvmean-predict(fitAparch))^2)
qlikeAparch=mean(log(predict(fitAparch))+rvmean/predict(fitAparch))

mseAll=mean((rvmean-predict(fitAll))^2)
qlikeAll=mean(log(predict(fitAll))+rvmean/predict(fitAll))

mseAllVix=mean((rvmean-predict(fitAllVix))^2)
qlikeAllVix=mean(log(predict(fitAllVix))+rvmean/predict(fitAllVix))

mseApfam=mean((rvmean-predict(fitApfam))^2)
qlikeApfam=mean(log(predict(fitApfam))+rvmean/predict(fitApfam))

mseArVvix=mean((rvmean-predict(fitArVvix))^2)
qlikeArVvix=mean(log(predict(fitArVvix))+rvmean/predict(fitArVvix))

mseApfamVvix=mean((rvmean-predict(fitApfamVvix))^2)
qlikeApfamVvix=mean(log(predict(fitApfamVvix))+rvmean/predict(fitApfamVvix))

mseaaa=mean((rvmean-predict(fitaaa))^2)
qlikeaaa=mean(log(predict(fitaaa))+rvmean/predict(fitaaa))

## calcolo dei migliori MSE e QLIKE

min.mse<-min(c(mseArma,mseGarch,mseTsgarch,mseVvix,
mseAparch, mseAll,
           mseAllVix,mseApfam,mseArVvix,mseApfamVvix))

cat("\n",

```

```

      "MSE ARMA", mseArma, "\n",
      "MSE GARCH", mseGarch, "\n",
      "MSE TSGARCH", mseTsgarch,
"\n",
      "MSE VIX", mseVvix, "\n",
      "MSE APARCH", mseAparch,
"\n",
      "MSE ALL", mseAll, "\n",
      "MSE ALL+VIX", mseAllVix, "\n",
"MSE APARCH+GARCH+TSGARCH", mseApfam, "\n",
"MSE ARMA+VIX", mseArVvix, "\n",
"MSE APARCH+GARCH+TSGARCH+VIX", mseApfamVvix, "\n")

```

```
min.mse
```

```
min.mse<-min(c(mseArma,mseArfima,mseGarch,mseTsgarch,mseVvix,
mseAparch))
```

```
min.qlike<-min(c(qlikeArma,qlikeGarch,qlikeTsgarch,qlikeVvix,
qlikeAparch, qlikeAll,
```

```
qlikeApfam,qlikeArVvix,qlikeApfamVvix, qlikeAllVix))
```

```
cat("\n",
```

```

      "QLIKE ARMA", qlikeArma, "\n",
      "QLIKE GARCH", qlikeGarch,
"\n",
      "QLIKE TSGARCH", qlikeTsgarch,
"\n",
      "QLIKE VIX", qlikeVvix, "\n",
      "QLIKE APARCH", qlikeAparch,
"\n",
      "QLIKE ALL", qlikeAll,
"\n",

```

```

"QLIKE ALL+VIX          ",
qlikeAllVix, "\n",

"QLIKE APARCH+GARCH+TSGARCH      ", qlikeApfam, "\n",

"QLIKE ARMA+VIX          ", qlikeArVvix, "\n",

"QLIKE APARCH+GARCH+TSGARCH+VIX  ",
qlikeApfamVvix, "\n")

```

```
min.qlike
```

```
min.qlike<-min(c(qlikeArma,qlikeGarch,qlikeTsgarch,qlikeVvix,
qlikeAparch))
```

```
##### DM TEST #####
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitTsgarch)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitArma)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitAparch)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitArVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitApfam)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitAllVix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitApfamVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitAll)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitGarch)),(rvmean-
predict(fitaaa)), alternative="greater", h=22)
```

```
#####
```

```
dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitGarch))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitArma))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitAparch))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitApfam))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitAllVix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitApfamVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitAll))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitTsgarch))),((rvmean-
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitGarch))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitTsgarch))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitAparch))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitApfam))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitAllVix))), alternative="greater", h=22)

dm.test(((rvmean-predict(fitArma))),((rvmean-
predict(fitApfamVvix))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArma)),(rvmean-  
predict(fitAll)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArma)),(rvmean-  
predict(fitaaa)), alternative="greater", h=22)
```

```
####
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitGarch)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitTsgarch)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitArma)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitArVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitApfam)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitAllVix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitApfamVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitAll))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitAparch)),(rvmean-  
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitGarch)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitTsgarch))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitArma))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitAparch))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitArVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitApfam)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitAllVix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitApfamVvix)), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitAll))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitVvix)),(rvmean-  
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitGarch))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitTsgarch))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitArma))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitAparch))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitVvix))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitApfam))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitAllVix))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitApfamVvix))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitAll))), alternative="greater", h=22)
```

```
dm.test((rvmean-predict(fitArVvix)),(rvmean-  
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitGarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitTsgarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitArma))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitAparch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitAllVix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitApfamVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitAll))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitApfam))), ((rvmean-
predict (fitaaa))), alternative="greater", h=22)

#####

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitGarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitTsgarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitArma))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitAparch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitAllVix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAll))), ((rvmean-
predict (fitApfamVvix))), alternative="greater", h=22)

```

```
dm.test(((rvmean-predict(fitAll))), ((rvmean-  
predict(fitApfam))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitAll))), ((rvmean-  
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitGarch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitTsgarch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitArma))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitAparch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitVvix))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitArVvix))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitApfam))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitAll))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitAllVix))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitApfamVvix))), ((rvmean-  
predict(fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test(((rvmean-predict(fitAllVix))), ((rvmean-  
predict(fitGarch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitAllVix))), ((rvmean-  
predict(fitTsgarch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitAllVix))), ((rvmean-  
predict(fitArma))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict(fitAllVix))), ((rvmean-  
predict(fitAparch))), alternative="greater", h=22)
```

```
dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitApfam))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitAll))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitApfamVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitAllVix))), ((rvmean-
predict (fitaaa))), alternative="greater", h=22)
```

```
#####
```

```
dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitGarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitTsgarch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitArma))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitAparch))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitArVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitApfam))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitAll))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitApfamVvix))), alternative="greater", h=22)

dm.test(((rvmean-predict (fitaaa))), ((rvmean-
predict (fitAllVix))), alternative="greater", h=22)
```

```
##### AIC combinazioni #####
```

```
aic.arma=(glm(rvmean~I (exp (2*arma))))$aic
```

```

aic.garch=(glm(rvmean~garch))$aic
aic.tsgarch=(glm(rvmean~tsgarch))$aic
aic.aparch=(glm(rvmean~aparch))$aic
aic.vvix=(glm(rvmean~vvix))$aic
aic.AllVvix=(glm(rvmean~I(exp(2*arma))+garch+tsgarch+aparch+vvix))$aic
aic.All=(glm(rvmean~I(exp(2*arma))+garch+tsgarch+aparch))$aic
aic.Apfam=(glm(rvmean~garch+aparch+tsgarch))$aic
ai.ArVvix=(glm(rvmean~I(exp(2*arma))+vvix))$aic
aic.ApfamVvix=(glm(rvmean~garch+aparch+tsgarch+vvix))$aic

min.aiclml<-min(c(aic.arma,aic.garch,aic.tsgarch,aic.vvix,
aic.aparch, aic.All,
                aic.AllVvix,aic.Apfam,aic.ArVvix,aic.ApfamVvix))

cat("\n",
    "AIC ARMA",
    "\n",
    "AIC TSGARCH",
    "\n",
    "AIC VIX",
    "\n",
    "AIC APARCH",
    "\n",
    "AIC ALL",
    "AIC ALL+VIX",
    aic.AllVvix,"\\n",
    "AIC APARCH+GARCH+TSGARCH",
    "AIC ARMA+VIX",
    "AIC APARCH+GARCH+TSGARCH+VIX",
    aic.ApfamVvix,"\\n")

```

min.aiclm

Bibliografia

[1] Becker R. Clements A. E. (2008): *Are combination forecasts of S&P 500 volatility statistically superior?*. International journal of forecasting

[2] Tasy R.S. (2002). *Analysis of financial time series*, Wiley, New York

[3] Ser-Huang Poon Clive W. J. Granger (2003). *Forecasting Volatility in Financial Markets: A Review*. Journal of Economic Literature

[4] D'Elia A. Piccolo D. (2002). *Stimatori di minima distanza del parametro alle differenze frazionarie*. Quaderni di Statistica Vol.4, 2002

[5] Corduas M. Piccolo D. (2002). *Una verifica della modellistica a differenza frazionaria per la serie delle portate del Tevere*. Quaderni di Statistica Vol. 4, 2002

[6] Bordignon S. (2008). Dispense Serie storiche finanziarie

<http://www.statistica.unipd.it/servizi/matdid.asp?idins=162#corso>

Ringraziamenti

È giunto il momento di fare qualche sentito ringraziamento.

Ringrazio innanzitutto il professor Silvano Bordignon per la sua immensa disponibilità e pazienza per i consigli e per i venerdì pomeriggio passati nel suo studio a risolvermi i problemi.

Un ringraziamento ai miei genitori e a Paola che hanno, quasi, pazientemente sopportato il mio stato di mutismo nervoso durante la stesura della tesi e nei periodi d'esame. Non dovevo far scappare via le poche parole che ho.

Per ultimi ma non meno importanti, ai miei colleghi e amici che mi hanno fatto passare 3 meravigliosi anni di vita universitaria.