Università degli Studi di Pisa
Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

Progetto di Statistica II

- Dipartimento di Ingegneria Gestionale -
Docente il *Professor Marco Romito*

Marco Tarsia
mtarsia@student.dm.unipi.it

Dicembre 2016

Introduzione

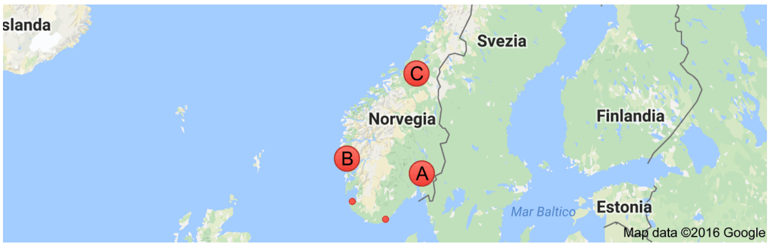
Nello svolgere questo lavoro possiamo supporre, con un piccolo sforzo di fantasia, di far attivamente parte di un gruppo internazionale di matematici statistici al quale è stato da poco commissionato un ben preciso compito da parte dell'azienda multinazionale Ikea, che immaginiamo ormai abbastanza intenzionata ad espandersi ulteriormente nella penisola scandinava ed in particolare in Norvegia.

Tale compito è quello di condurre un'analisi giustappunto statistica di opportuni dati numerici ufficiali norvegesi la quale possa contribuire almeno parzialmente alla decisione di aprire o meno un nuovo store di dimensioni relativamente contenute fra la grande città centrale di Trondheim - di circa 180000 abitanti - e la città di Mosjøen - di circa 10000 abitanti -, più a nord-est.

Dunque possiamo senz'altro cominciare col procurarci in rete e col metterci subito da parte una qualsiasi cartina politica della Norvegia:

Adesso, sempre grazie a rapide quanto innocue ricerche in rete, possiamo conoscere la distribuzione dei principali stores di Ikea nel paese nordico ed osservare intanto che, in effetti, lo stabilimento attualmente più a nord è l'*Ikea Leangen* di Trondheim, come possiamo vedere chiaramente grazie ad un'altra cartina della Norvegia, o meglio della sola parte di paese circa al di sotto proprio di Mosjøen, nella quale in particolare tale stabilimento viene indicato con la lettera maiuscola "C".

Teniamo da parte pure tale cartina ed associamole inoltre un'elementare tabella di dati basilari specifici degli stores segnalati - di nuovo ufficiali e facilmente rintracciabili in rete - compilata seguendo la precisa convenzione di descriverli scendendo da nord verso sud:



Store	**Città**	**Abitanti** (circa)	**Apertura**	$\mathbf{m^2}$ (circa)
C	Trondheim	180000	2002	22500
B	Nyborg	17000	1984	12600
A	Oslo	620000	1998	20500
A	Billingstadm, Oslo	3000	1963	33500
altro	Sandnes	75000	1988	18100
altro	Høvåg	-	-	-

Ragionando senza troppa fretta sulla situazione fin ora presentata, non verrebbe da escludere subito il fatto che davvero possa aver senso l'apertura a nord-est di Trondheim di un nuovo stabilimento, il quale però presenti magari un'estensione non superiore ai 16000/18000 m^2 circa.

Pertanto, a questo punto, possiamo veramente far entrare in gioco i vari dati statistici ufficiali norvegesi che più rigorosamente possano aiutare per una tale scelta aziendale; fra i maggiormente significativi in questo senso, ad esempio, quelli relativi a: le caratteristiche economiche "generali" del paese, lo sviluppo industriale specialmente manifatturiero, i costi dell'energia, le tasse sulle proprietà, la crescita demografica della popolazione, i fenomeni di emigrazione e di immigrazione, la tipologia ed il benessere economico della popolazione, l'andamento delle unioni matrimoniali e civili, i mezzi di trasporto propri ed il traffico in generale, le temperature e le precipitazioni, il tasso di disoccupazione specialmente giovanile, i prezzi delle case, le spese nell'arredamento di interni generici.

Ebbene, possiamo pensare che il gruppo di studio del quale facciamo felicemente parte debba appunto occuparsi perlomeno di tale insieme di fattori, od anche eventualmente di un qualche sondaggio appositamente organizzato, e che soprattutto il nostro lavoro debba realizzarsi sui soli ultimi tre di quelli or ora elencati, per lo più rispetto al passare del tempo; esplicitamente: tasso di disoccupazione specialmente giovanile nel tempo, prezzi delle case nel tempo, e spese nell'arredamento di interni generici nel tempo.

Caliamoci quindi nella nostra parte senza ulteriori indugi iniziando col procurarci dal sito ufficiale di statistiche norvegesi `https://www.ssb.no/en` tre *serie storiche* della specie in questione, ed analizziamole in dettaglio una ad una servendoci del noto software **R** su sistema operativo Mac OS X per poi proporre da ultima una possibile sintesi finale del tutto.

1 La prima serie storica: tasso di disoccupazione

Delle serie relative al tasso di disoccupazione specialmente giovanile - tutte ufficialmente disponibili ***qui***, per l'esattezza - decidiamo di studiare la seguente, intendendola come quella "canonicamente" associata alla tabella che proponiamo subito dopo:

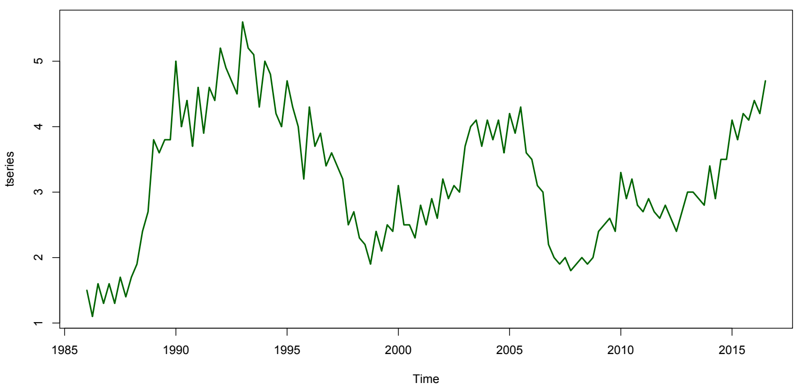
1.5 1.1 1.6 1.3 1.6 1.3 1.7 1.4 1.7 1.9 2.4 2.7 3.8 3.6 3.8 3.8 5 4 4.4 3.7 4.6 3.9 4.6 4.4 5.2 4.9 4.7 4.5 5.6 5.2 5.1 4.3 5 4.8 4.2 4 4.7 4.3 4 3.2 4.3 3.7 3.9 3.4 3.6 3.4 3.2 2.5 2.7 2.3 2.2 1.9 2.4 2.1 2.5 2.4 3.1 2.5 2.5 2.3 2.8 2.5 2.9 2.6 3.2 2.9 3.1 3 3.7 4 4.1 3.7 4.1 3.8 4.1 3.6 4.2 3.9 4.3 3.6 3.5 3.1 3 2.2 2 1.9 2 1.8 1.9 2 1.9 2 2.4 2.5 2.6 2.4 3.3 2.9 3.2 2.8 2.7 2.9 2.7 2.6 2.8 2.6 2.4 2.7 3 3 2.9 2.8 3.4 2.9 3.5 3.5 4.1 3.8 4.2 4.1 4.4 4.2 4.7 .

Anno	Trim. 1	Trim. 2	Trim. 3	Trim. 4
1986	1.5	1.1	1.6	1.3
1987	1.6	1.3	1.7	1.4
1988	1.7	1.9	2.4	2.7
1989	3.8	3.6	3.8	3.8
1990	5	4	4.4	3.7
1991	4.6	3.9	4.6	4.4
1992	5.2	4.9	4.7	4.5
1993	5.6	5.2	5.1	4.3
1994	5	4.8	4.2	4
1995	4.7	4.3	4	3.2
1996	4.3	3.7	3.9	3.4
1997	3.6	3.4	3.2	2.5
1998	2.7	2.3	2.2	1.9
1999	2.4	2.1	2.5	2.4
2000	3.1	2.5	2.5	2.3
2001	2.8	2.5	2.9	2.6
2002	3.2	2.9	3.1	3
2003	3.7	4	4.1	3.7
2004	4.1	3.8	4.1	3.6
2005	4.2	3.9	4.3	3.6
2006	3.5	3.1	3	2.2
2007	2	1.9	2	1.8
2008	1.9	2	1.9	2
2009	2.4	2.5	2.6	2.4
2010	3.3	2.9	3.2	2.8
2011	2.7	2.9	2.7	2.6
2012	2.8	2.6	2.4	2.7
2013	3	3	2.9	2.8
2014	3.4	2.9	3.5	3.5
2015	4.1	3.8	4.2	4.1
2016	4.4	4.2	4.7	NA

Da una parte abbiamo così una serie già pronta per esser agilmente caricata su **R**, il ché non può minimamente dispiacerci; inoltre, d'altra parte, abbiamo pure la sua forma più completa e chiara della quale conviene in effetti restar sempre ben consapevoli - già a cominciare dalla scelta dei due parametri di partenza `frequency` e `start` del codice.

Evidenziamo dunque tale serie, copiamola mediante ad esempio l'usuale combinazione di tasti `cmd+c`, e procediamo finalmente su **R** coi primissimi comandi ed il primissimo grafico a riguardo:

```
> X <- scan(pipe("pbpaste"))
Read 123 items
> tseries = ts(X,frequency=4,start=c(1986,1))
> ts.plot(tseries,lwd=2,col="darkgreen")
```



La serie così rappresentata, nonostante sembri alquanto complicata da rumore di una certa importanza non trascurabile, parrebbe mostrare in netta prevalenza una struttura di trend apparentemente più ascendente che non, specie in corrispondenza degli ultimi sedici anni - i quali devono forse esser per noi di particolare interesse in ogni caso -, a confronto invece di una stagionalità oggettivamente trascurabile, con la possibilità che esista piuttosto una *ciclicità* di fondo la quale tenda a svilupparsi in modo quasi stazionario, o la quale magari venga per così dire via via smorzata - viste infatti le tre "gobbe" del grafico, sempre più piccole, prima della salita finale - per lasciare che un'eventuale tendenza al trend ascendente abbia infine il sopravvento.

In effetti, non soltanto a livello intuitivo non ci aspetteremmo una vera stagionalità quanto semmai una ciclicità, appunto; ma soprattutto il fatto che ci sia davvero o meno dovrebbe risultare pressoché ininfluente rispetto a ciò che più c'interessa comprendere del comportamento strutturale della serie in questione rispetto ai nostri fini, ovvero naturalmente proprio il suo trend.

Questo potrebbe in particolare rassicurarci un po' sulla nostra scelta iniziale del parametro `frequency`, ma resta comunque ovvia la necessità di passare a vere analisi matematiche tecniche a riguardo.

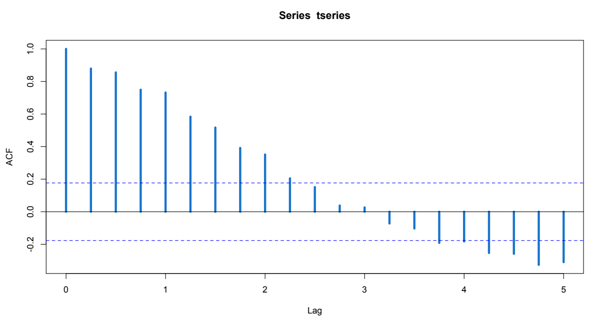
1.1 Funzione di auto-correlazione, decomposizioni della serie e analisi dei residui

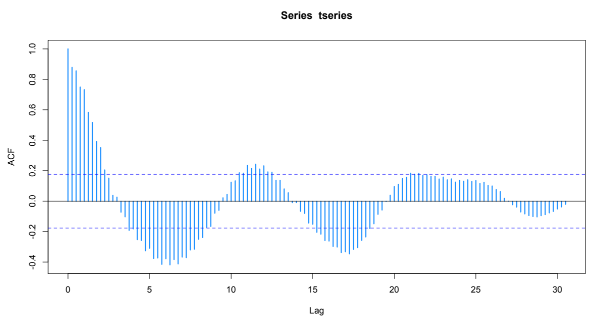
Decidiamo di lavorare sulla serie per intero e di basarci per ciò sui comandi digitati poco fa. Anzi, rimarchiamo subito ed una volta per tutte che in generale ogni parte in codice del nostro progetto sarà una naturale e comunque evidente continuazione di comandi quasi immediatamente precedenti.

1.1.1 Funzione di auto-correlazione

Partiamo con la ACF su due lag estremali:

```
> acf(tseries,lwd=4,col="dodgerblue3")
> acf(tseries,125,lwd=2,col="dodgerblue")
```



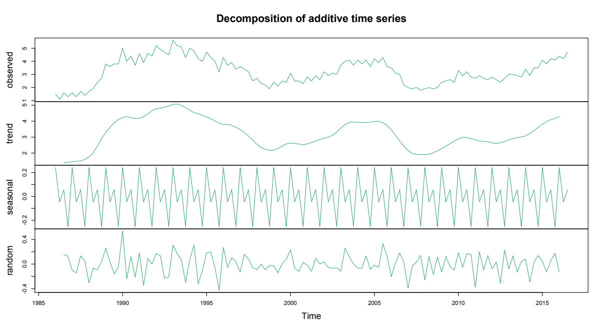


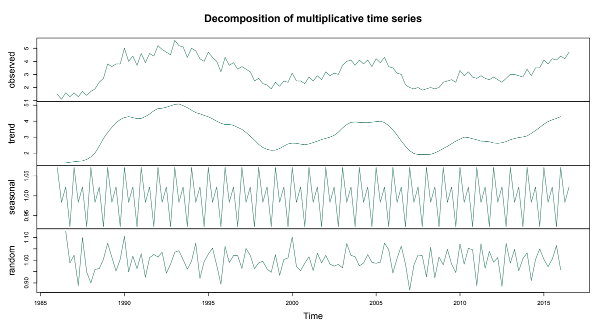
Tutto in linea con le nostre aspettative, sebbene un `frequency` di circa `6` sembri più appropriato alla situazione. Siamo tuttavia piuttosto tranquilli nel provare per ora a procedere così.

1.1.2 Decomposizioni della serie

Decomposizione standard additiva e decomposizione moltiplicativa rispettivamente:

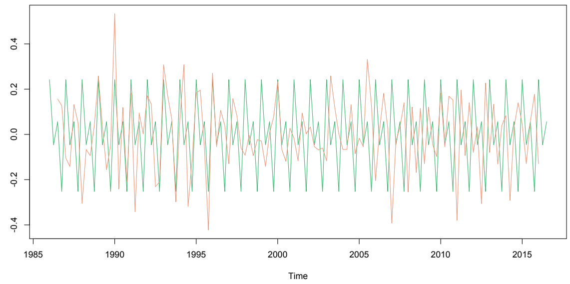
```
> dec.add = decompose(tseries)
> dec.mult = decompose(tseries,type="multiplicative")
> plot(dec.add,col="mediumseagreen")
> plot(dec.mult,col="aquamarine4")
```

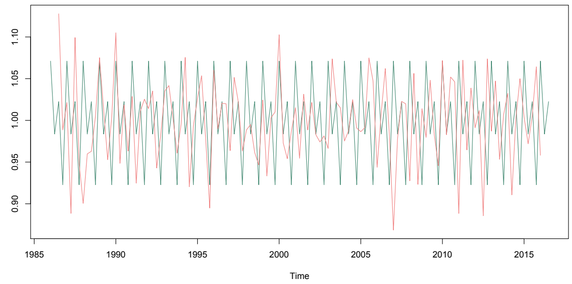




Dunque, in effetti, entrambe le decomposizioni rivelano una stagionalità ed un rumore di un medesimo ordine di grandezza che risulta decisamente inferiore rispetto a quello del trend. Vediamolo più chiaramente: da una parte,

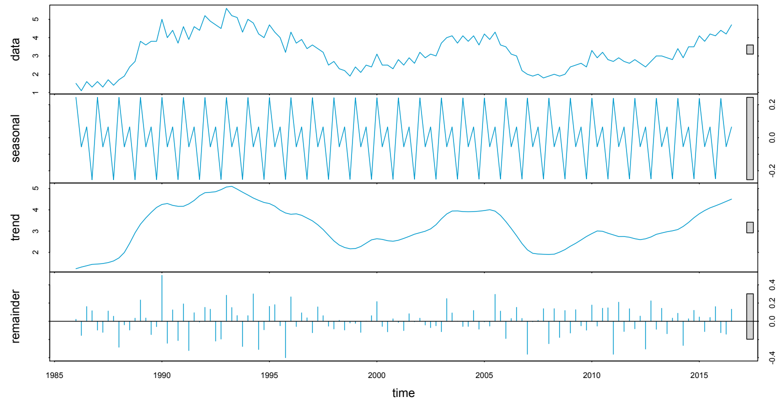
```
> ts.plot(dec.add$seasonal,dec.add$random,col=c("mediumseagreen","darksalmon"))
> ts.plot(dec.mult$seasonal,dec.mult$random,col=c("aquamarine4","lightcoral"))
```





D'altra parte,

```
> plot(stl(tseries,125),col="deepskyblue3")
```



In definitiva, la struttura della serie è davvero di trend dominante: manteniamo pertanto il nostro `frequency=4` e cerchiamo di capire quale delle due decomposizioni la "catturi meglio", senza indagare invece quanto offrirebbero degli `stl` più mirati.

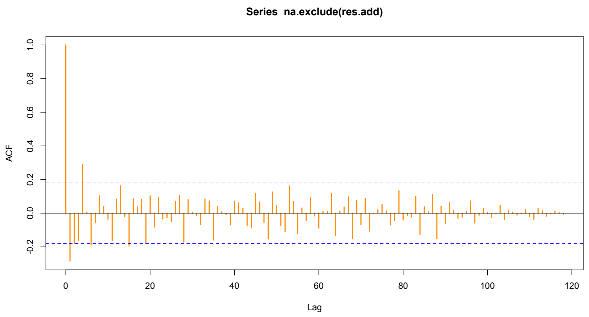
1.1.3 Analisi dei residui

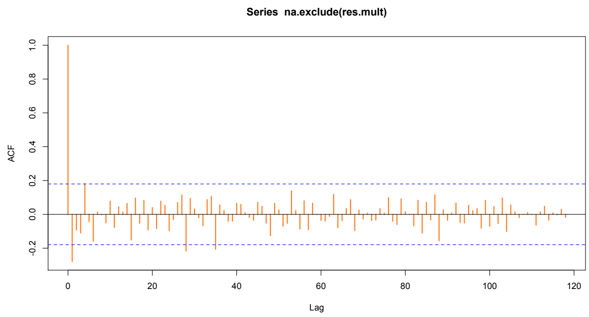
Rendiamo subito omogenee le due serie dei residui e confrontiamole poi in modo il più possibile esauriente:

```
> res.add = dec.add$random
> res.mult = dec.mult$random
> trend.mult = dec.mult$trend
> res.mult = mean(na.exclude(trend.mult)) * (res.mult - 1)
```

Rispettive funzioni di auto-correlazione integrali:

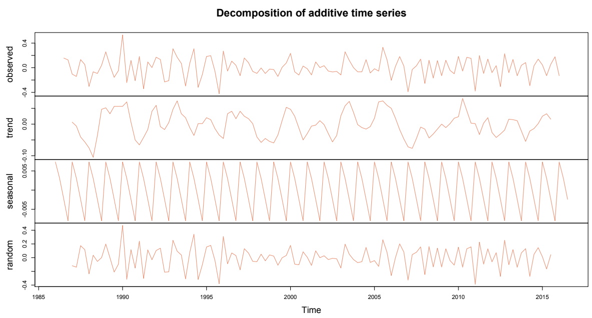
```
> acf(na.exclude(res.add),125,lwd=2,col="darkorange")
> acf(na.exclude(res.mult),125,lwd=2,col="chocolate1")
```

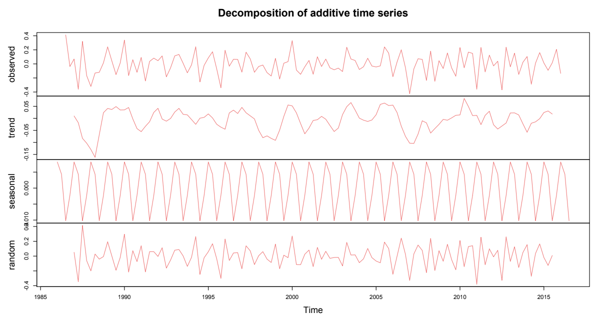




Rispettive decomposizioni standard additive:

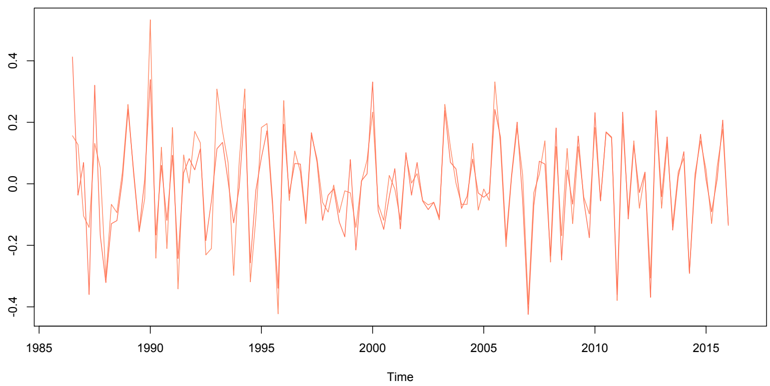
```
> plot(decompose(res.add),col="darksalmon")
> plot(decompose(res.mult),col="lightcoral")
```





Le due serie semplicemente disegnate assieme:

```
> ts.plot(res.add,res.mult,col=c("salmon1","coral1"))
```



Rispettive varianze spiegate calcolate tenendo conto dei rispettivi dati `NA` presenti:

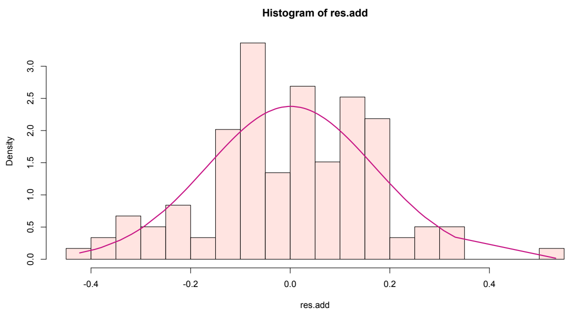
```
> 1 - var(na.exclude(res.add))/var(tseries[3:121])
[1] 0.9711552
> 1 - var(na.exclude(res.mult))/var(tseries[3:121])
[1] 0.9732387
```

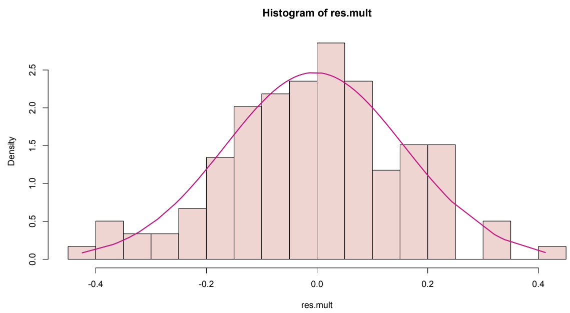
A proposito, una piccola curiosità a riguardo della seconda serie dei residui:

```
> 1 - var(na.exclude(log(dec.mult$random)))/var(tseries[3:121])
[1] 0.9973924
```

Rispettive medie empiriche e quindi rispettivi istogrammi - i quali naturalmente non siano troppo fini:

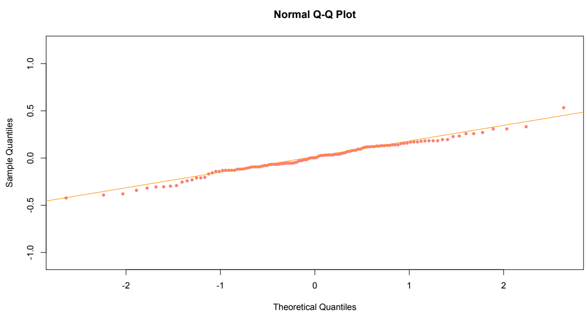
```
> res.add.s = sort(na.exclude(res.add))
> Z_1 = dnorm(res.add.s,mean(res.add.s),sd(res.add.s))
> mean(res.add.s)
[1] 0.0009806034
> res.mult.s = sort(na.exclude(res.mult))
> Z_2 = dnorm(res.mult.s,mean(res.mult.s),sd(res.mult.s))
> mean(res.mult.s)
[1] -0.004443235
> hist(res.add,15,freq=FALSE,col="mistyrose1")
> lines(res.add.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.mult,15,freq=FALSE,col="mistyrose2")
> lines(res.mult.s,Z_2,lwd=2,col="mediumvioletred")
```

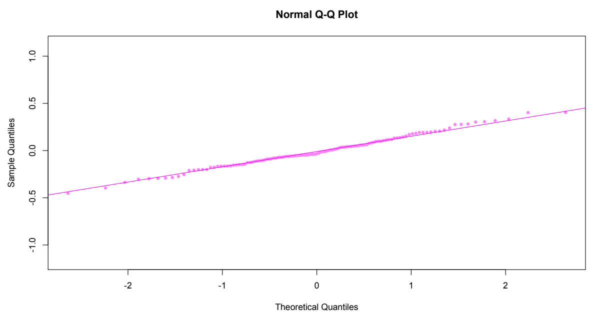




Rispettivi quantili, finalmente: per la prima,

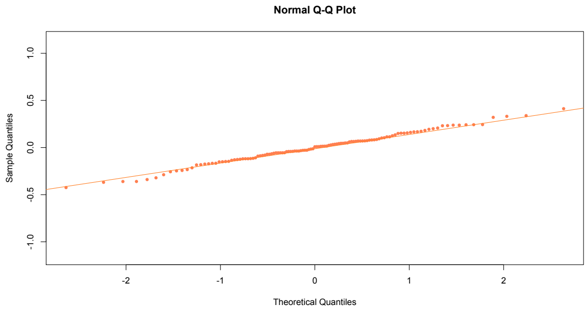
```
> qqnorm(res.add.s,pch=20,col="salmon",asp=1)
> qqline(res.add.s,col="darkorange")
> Z_1 = rnorm(length(res.add.s),mean(res.add.s),sd(res.add.s))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```

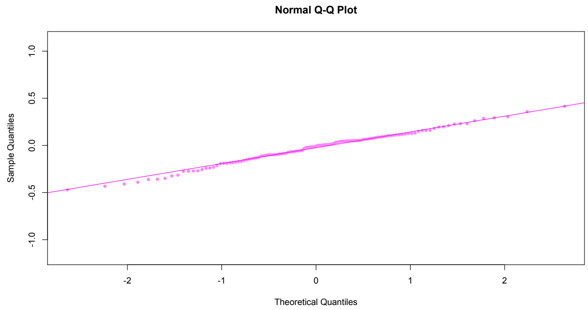




Per la seconda,

```
> qqnorm(res.mult.s,pch=20,col="coral",asp=1)
> qqline(res.mult.s,col="chocolate1")
> Z_2 = rnorm(length(res.mult.s),mean(res.mult.s),sd(res.mult.s))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```





Visto tutto, concludiamo che le due decomposizioni sono molto simili e davvero ben riuscite, e dell'analisi dei relativi residui apprezziamo specialmente le varianze spiegate ed i `qqnorm` ammettendo viceversa che le due funzioni di auto-correlazione potevano esser leggermente più soddisfacenti.

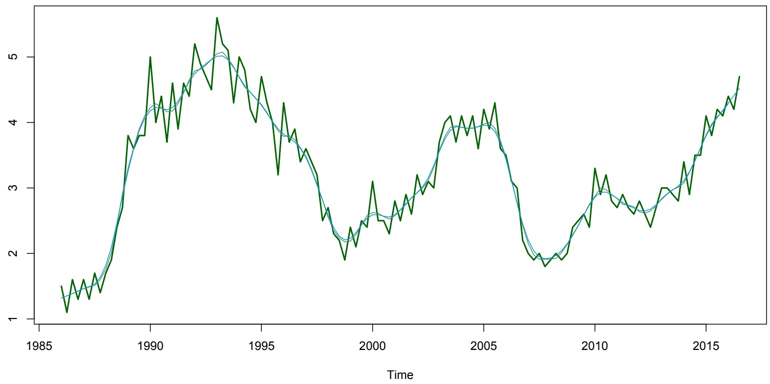
Dovessimo a questo punto proporre lo stesso come "migliore" una delle due decomposizioni basandosi sulle analisi condotte, sceglieremmo probabilmente la seconda o cioè quella moltiplicativa.

1.2 Analisi e previsioni del trend e della serie

Grazie a quanto fin ora compreso, consideriamo opportuna una previsione dell'andamento della serie in un prossimo futuro - nell'anno 2017, praticamente - la quale sia il risultato del confronto di analisi e previsione del solo trend con analisi e previsione di un filtraggio della serie il quale sia una specie di via di mezzo proprio tra trend e serie stessa.

Cominciamo con calma tracciando insieme la serie, il suo trend ed anche uno smoothing del medesimo per poi analizzarli e quindi prevederli separatamente uno ad uno, convenendo naturalmente di esser ben più accurati nel farlo per l'intera serie:

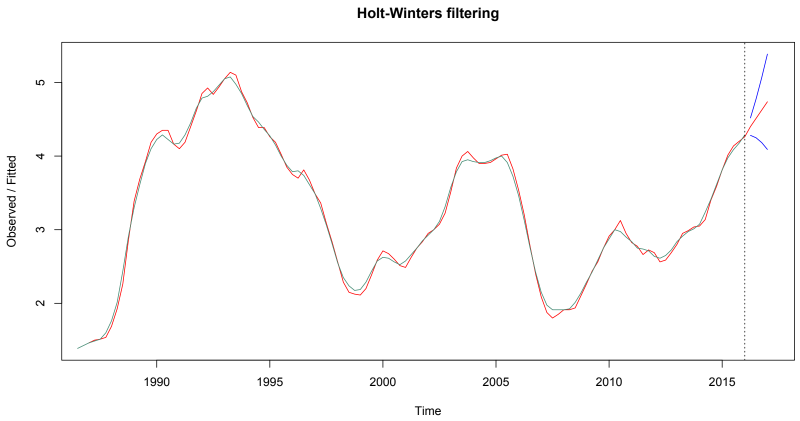
```
> ts.plot(tseries,trend.mult,stl(tseries,5)$time.series[,2],lwd=c(2,1,1),
+ col=c("darkgreen","aquamarine4","deepskyblue3"))
```



1.2.1 Analisi e previsioni del trend

Partiamo dallo smorzamento esponenziale con trend (SET) automatico per il trend:

```
> trend = trend.mult
> trend.SET = HoltWinters(na.omit(trend),gamma=FALSE)
> plot(trend.SET,predict(trend.SET,4,prediction.interval=TRUE),col="aquamarine4")
```

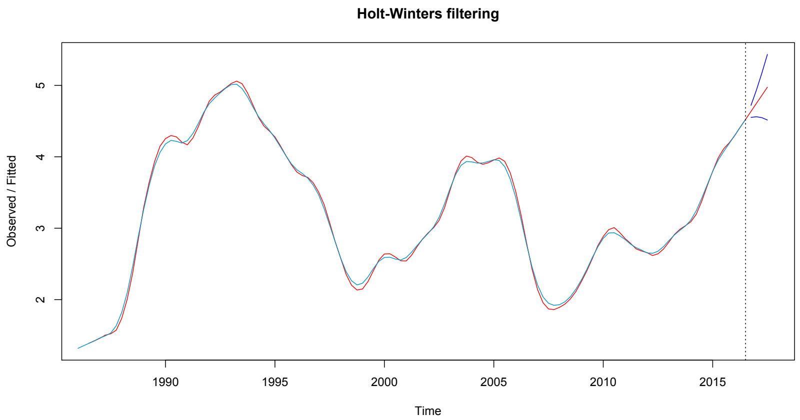


Notiamo una semplice riprova della sua efficacia:

```
> trend.SET

Smoothing parameters:
 alpha: 1
 beta : 1
 gamma: FALSE
```

Guardiamo quindi il SET automatico per lo smoothing del trend:

```
> trend = stl(tseries,5)$time.series[,2]
> trend.SET = HoltWinters(na.omit(trend),gamma=FALSE)
> plot(trend.SET,predict(trend.SET,4,prediction.interval=TRUE),col="deepskyblue3")
```

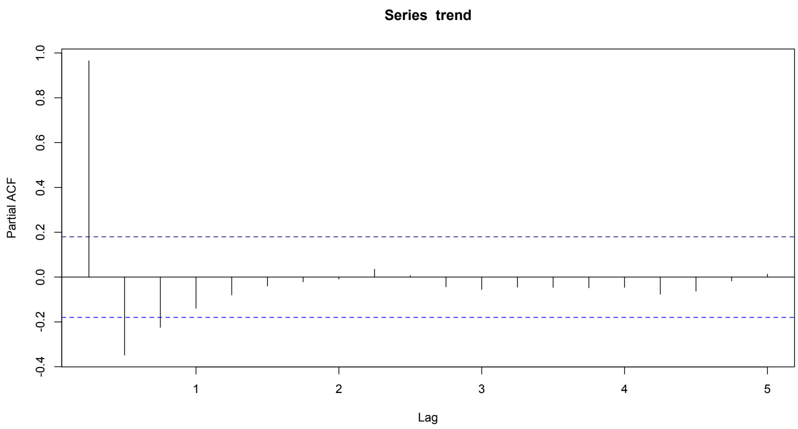


Medesima riprova:

```
> trend.SET

Smoothing parameters:
 alpha: 1
 beta : 1
 gamma: FALSE
```

Procediamo adesso con un'auto-regressione lineare effettuata manualmente per il trend, basandosi sulla funzione di auto-correlazione parziale per il trend stesso:

```
> trend = na.omit(trend.mult)
> pacf(trend)
```



Costruiamo un modello lineare per il trend:

```
> n = length(trend)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(trend.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05401    0.02007   2.691 0.008298 **
A[, 1]       0.22750    0.09438   2.411 0.017684 *
A[, 2]      -0.83448    0.24688  -3.380 0.001021 **
A[, 3]       1.11608    0.31031   3.597 0.000495 ***
A[, 4]      -0.78218    0.31606  -2.475 0.014948 *
A[, 5]       0.99126    0.31063   3.191 0.001874 **
A[, 6]      -2.20797    0.24848  -8.886 2.05e-14 ***
A[, 7]       2.47399    0.09601  25.768  < 2e-16 ***

Residual standard error: 0.04507 on 104 degrees of freedom
Multiple R-squared:  0.9974,Adjusted R-squared:  0.9973
F-statistic:  5789 on 7 and 104 DF,  p-value: < 2.2e-16
```

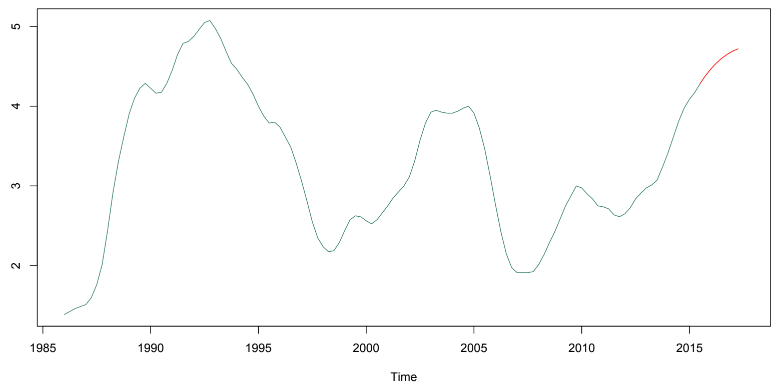
Tale modello possiede fra l'altro una varianza spiegata corretta R^2_{Adj} già ottima, ma lo ottimizziamo:

```
> trend.lm = lm(A[,8]~A[,6]+A[,7]+0)
> summary(trend.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 6] -0.92697    0.03649  -25.41   <2e-16 ***
A[, 7]  1.92553    0.03628   53.08   <2e-16 ***

Residual standard error: 0.05979 on 110 degrees of freedom
```

```
Multiple R-squared:  0.9997,Adjusted R-squared:  0.9997
F-statistic: 1.857e+05 on 2 and 110 DF,  p-value: < 2.2e-16
```

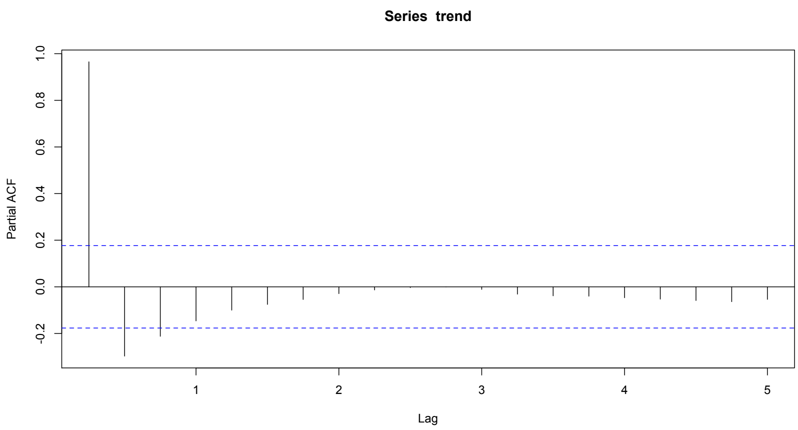
Conseguente previsione per il trend:

```
> pred = 1:(n+k-1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> trend.P = ts(trend.P,frequency=4,start=c(1986,1))
> pred.P = ts(pred.P,frequency=4,start=c(1986,1))
> ts.plot(trend.P,pred.P,col=c("aquamarine4","red"))
```



Volendo ora far lo stesso per lo smoothing del trend, guardiamo la funzione di auto-correlazione parziale per tale smoothing:

```
> trend = na.omit(stl(tseries,5)$time.series[,2])
> pacf(trend)
```



Costruiamo un modello lineare per lo smoothing:

```
> n = length(trend)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(trend.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.017275   0.006599   2.618   0.0101 *
A[, 1]       0.425305   0.084905   5.009 2.15e-06 ***
A[, 2]      -1.627769   0.297401  -5.473 2.89e-07 ***
A[, 3]       2.744292   0.492788   5.569 1.89e-07 ***
A[, 4]      -3.377907   0.559697  -6.035 2.27e-08 ***
A[, 5]       4.386197   0.495810   8.847 1.93e-14 ***
A[, 6]      -4.989660   0.301067 -16.573  < 2e-16 ***
A[, 7]       3.434573   0.086665  39.630  < 2e-16 ***

Residual standard error: 0.01437 on 108 degrees of freedom
Multiple R-squared:  0.9997,Adjusted R-squared:  0.9997
F-statistic: 6.092e+04 on 7 and 108 DF,  p-value: < 2.2e-16
```

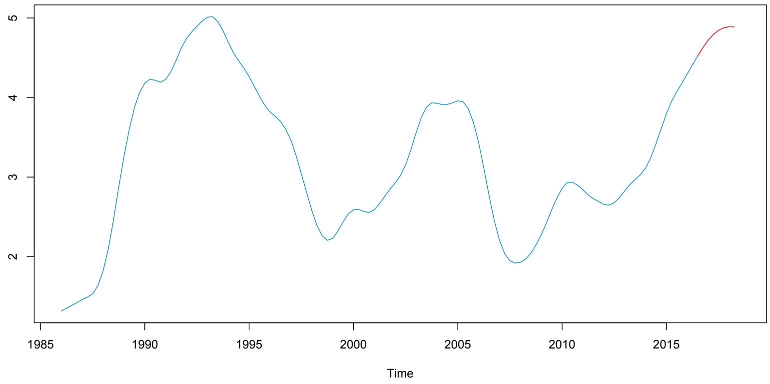
Tale modello possiede fra l'altro una varianza spiegata corretta R^2_{Adj} addirittura preferibile a quella di prima e praticamente perfetta in generale, ma lo possiamo ottimizzare facilmente:

```
> trend.lm = lm(A[,8]~A[,5]+A[,6]+A[,7]+0)
> summary(trend.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 5]  0.86842    0.04806   18.07   <2e-16 ***
A[, 6] -2.65188    0.09459  -28.04   <2e-16 ***
A[, 7]  2.78379    0.04769   58.37   <2e-16 ***

Residual standard error: 0.02204 on 113 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 9.409e+05 on 3 and 113 DF,  p-value: < 2.2e-16
```

Conseguente previsione per lo smoothing:

```
> pred = 1:(n+k-1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-3],pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
```

```
+ }
> trend.P = ts(trend.P,frequency=4,start=c(1986,1))
> pred.P = ts(pred.P,frequency=4,start=c(1986,1))
> ts.plot(trend.P,pred.P,col=c("deepskyblue3","red"))
```



Affianchiamo al tutto pure un metodo automatico dei minimi quadrati nell'auto-regressione lineare per il trend, e solo per questo:

```
> trend = na.omit(trend.mult)
> trend.ar = ar(trend,method="ols",aic=TRUE)
```

Una rapida occhiata alla varianza...

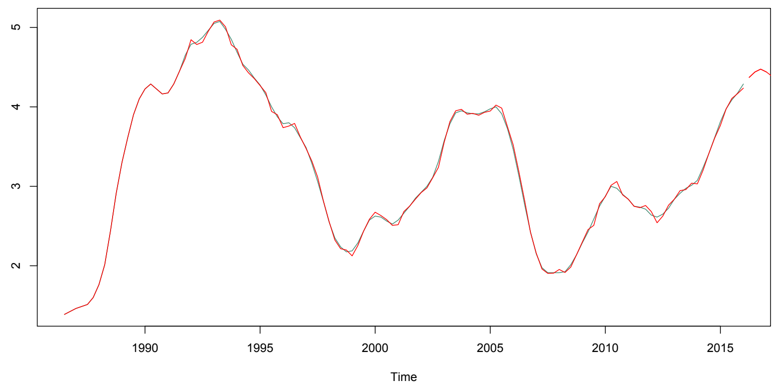
```
> trend.ar

Order selected 20  sigma^2 estimated as  0.001156
```

...e soprattutto alla varianza spiegata:

```
> 1 - trend.ar$var/var(trend)
          [,1]
[1,] 0.9986923
```

Analisi e previsione conseguenti al metodo:

```
> o = trend.ar$order
> a = trend.ar$ar
> trend.ar.P = trend
> for(i in (o+1):length(trend)){
+ trend.ar.P[i] = sum(rev(a)*trend[(i-o):(i-1)]) + mean(trend)*(1-sum(a))
+ }
> ts.plot(trend,trend.ar.P,col=c("aquamarine4","red"))
> lines(predict(trend.ar,n.ahead=8,se.fit=FALSE),col="red")
```



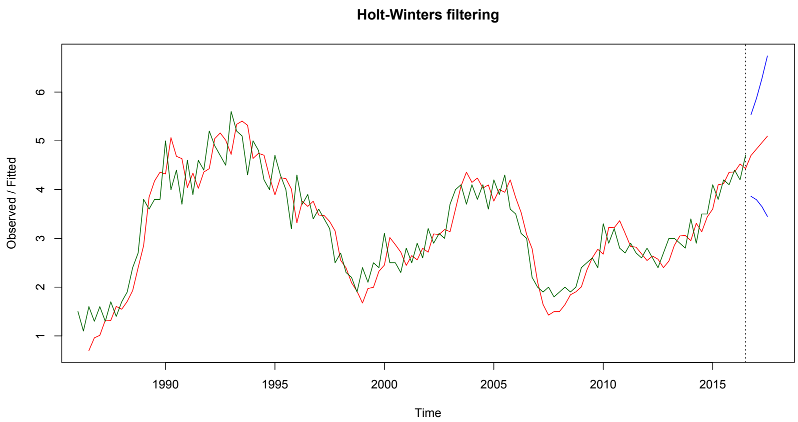
Dunque siamo alquanto soddisfatti dell'auto-regressione lineare svolta per il trend, e ancor più forse per lo smoothing scelto, in quanto sembra confermare non solo una tendenza del trend ad un'ascesa che sia piuttosto pronunciata ma che sia soprattutto temporanea e sottomessa ad una certa ciclicità di fondo nella quale tendiamo davvero a credere.

Di conseguenza ci sentiamo abbastanza fiduciosi in un'auto-regressione lineare anche per l'analisi e la previsione dell'intera serie, e speriamo in delle conclusioni analoghe.

1.2.2 Analisi e previsioni della serie

Per cominciare, proponiamo un SET automatico:

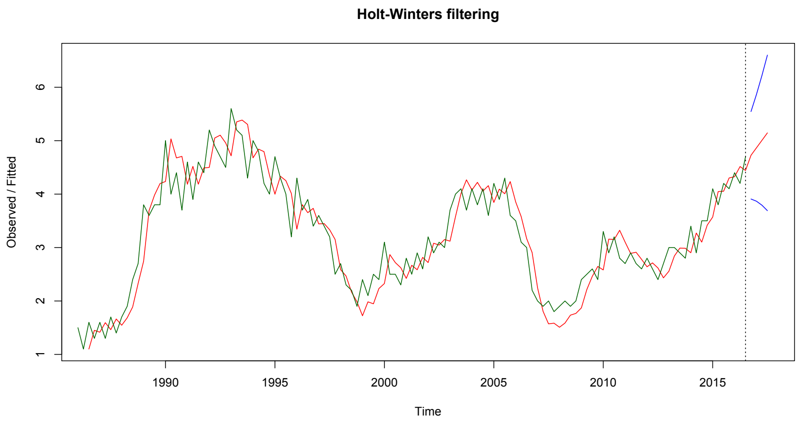
```
> tseries.SET = HoltWinters(tseries,gamma=FALSE)
> plot(tseries.SET,predict(tseries.SET,4,prediction.interval=TRUE),col="darkgreen")
> tseries.SET

Smoothing parameters:
 alpha: 0.5088538
 beta : 0.4424242
 gamma: FALSE
```



Inizializziamolo in modo più opportuno:

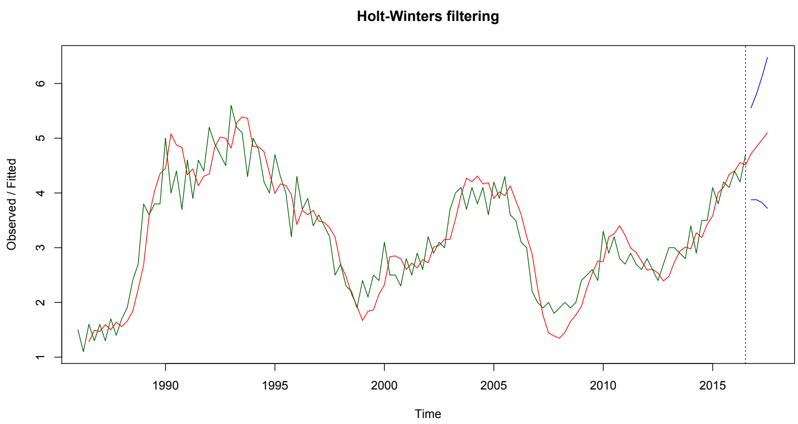
```
> tseries.SET = HoltWinters(tseries,gamma=FALSE,b.start=0)
> plot(tseries.SET,predict(tseries.SET,4,prediction.interval=TRUE),col="darkgreen")
> tseries.SET
```

```
Smoothing parameters:
 alpha: 0.5602482
 beta : 0.2513274
 gamma: FALSE
```



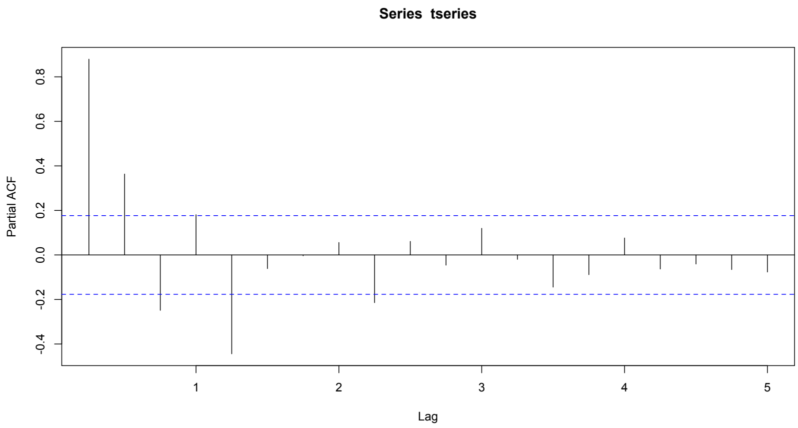
Optiamo tuttavia per il seguente SET calibrato manualmente, e facciamo questa scelta nella serena consapevolezza di basarci pesantemente su non pochi ragionamenti e tentativi affrontati a riguardo:

```
> tseries.SET = HoltWinters(tseries,alpha=0.4,beta=0.425,gamma=FALSE,
+ l.start=1.25,b.start=0.03)
> plot(tseries.SET,predict(tseries.SET,4,prediction.interval=TRUE),col="darkgreen")
```



Andiamo ora di auto-regressione lineare manuale partendo dalla funzione di auto-correlazione parziale:

```
> pacf(tseries)
```



Costruiamo un modello lineare:

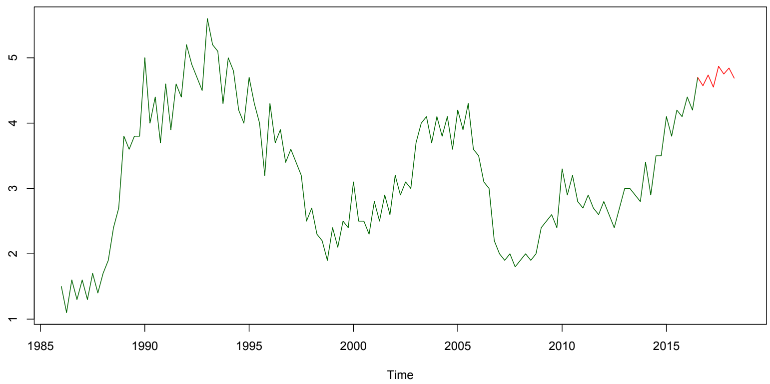
```
> n = length(tseries)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = tseries[j:(n-k+j)]
+ }
> tseries.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(tseries.lm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.35686    0.11705   3.049  0.00289 **
A[, 1]       0.08028    0.09484   0.846  0.39915
A[, 2]      -0.18200    0.11721  -1.553  0.12338
A[, 3]      -0.47739    0.11299  -4.225 5.01e-05 ***
A[, 4]       0.63708    0.09783   6.512 2.41e-09 ***
A[, 5]      -0.22979    0.11321  -2.030  0.04484 *
A[, 6]       0.33779    0.11780   2.868  0.00497 **
A[, 7]       0.73003    0.09623   7.586 1.23e-11 ***

Residual standard error: 0.3161 on 108 degrees of freedom
Multiple R-squared:  0.8945,Adjusted R-squared:  0.8877
F-statistic: 130.8 on 7 and 108 DF,  p-value: < 2.2e-16
```

Tale modello è già più che discreto, ma pure nettamente migliorabile:

```
> tseries.lm = lm(A[,8]~A[,3]+A[,4]+A[,7]+0)
> summary(tseries.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 3] -0.63394    0.07350  -8.625 4.56e-14 ***
A[, 4]  0.72085    0.07334   9.829  < 2e-16 ***
A[, 7]  0.91064    0.05229  17.416  < 2e-16 ***

Residual standard error: 0.3377 on 113 degrees of freedom
Multiple R-squared:  0.9907,Adjusted R-squared:  0.9904
F-statistic:  4009 on 3 and 113 DF,  p-value: < 2.2e-16
```

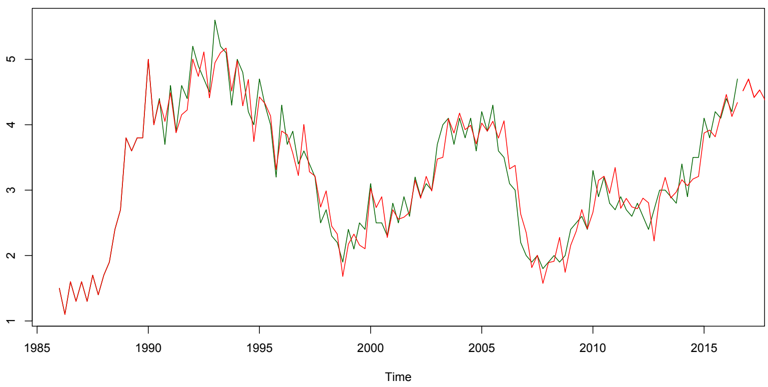
Conseguente previsione:

```
> pred = 1:(n+k-1)
> pred[1:n] = tseries
> for(i in 1:(k-1)){
+ pred[n+i] = coef(tseries.lm) %*% c(pred[n+i-5],pred[n+i-4],pred[n+i-1])
+ }
> tseries.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ tseries.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> tseries.P = ts(tseries.P,frequency=4,start=c(1986,1))
> pred.P = ts(pred.P,frequency=4,start=c(1986,1))
> ts.plot(tseries.P,pred.P,col=c("darkgreen","red"))
```



Occhiata al metodo automatico dei minimi quadrati nell'auto-regressione lineare:

```
> tseries.ar = ar(tseries,method="ols",aic=TRUE)

> tseries.ar

Order selected 18  sigma^2 estimated as  0.06668

> 1 - tseries.ar$var/var(tseries)
          [,1]
[1,] 0.9353976

> o = tseries.ar$order
> a = tseries.ar$ar
> tseries.ar.P = tseries
> for(i in (o+1):length(tseries)){
+ tseries.ar.P[i] = sum(rev(a)*tseries[(i-o):(i-1)]) + mean(tseries)*(1-sum(a))
+ }
> ts.plot(tseries,tseries.ar.P,col=c("darkgreen","red"))
> lines(predict(tseries.ar,n.ahead=8,se.fit=FALSE),col="red")
```



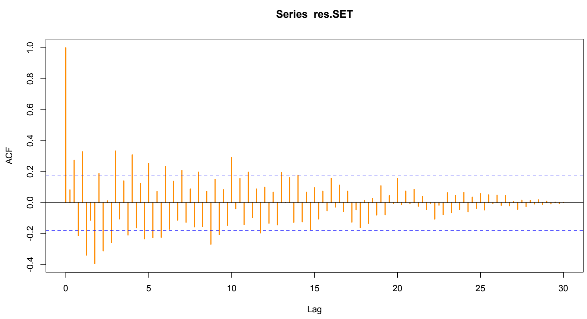
Così alla fine è tutto andato come credevamo e come speravamo: non resta ormai che trovare le ultime sicurezze nei numeri dati da un attento studio delle serie di residui.

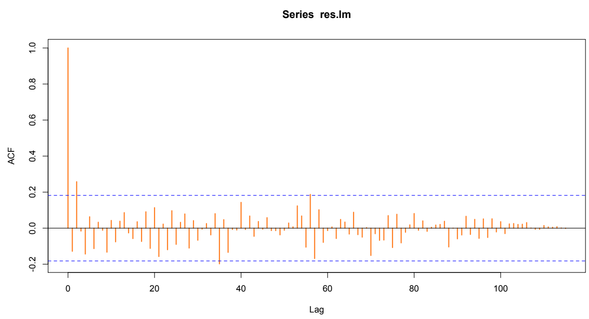
1.3 Analisi dei residui ed incertezza delle previsioni

1.3.1 Analisi dei residui

Estraiamo le due serie dei residui ed osserviamo le rispettive funzioni di auto-correlazione:

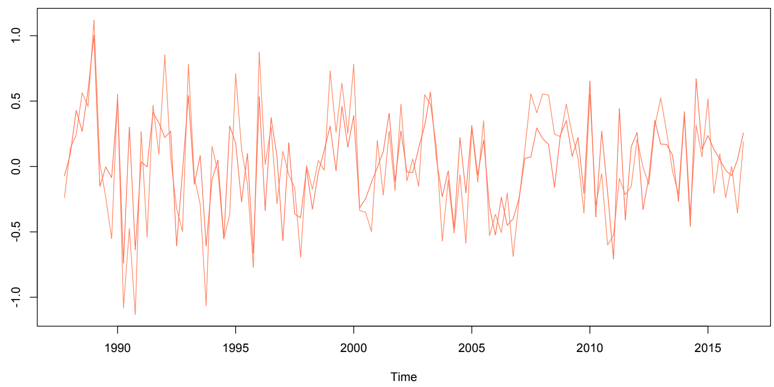
```
> res.SET = residuals(tseries.SET)
> length(res.SET)
[1] 121
> res.lm = residuals(tseries.lm)
> length(res.lm)
[1] 116
> acf(res.SET,125,lwd=2,col="darkorange")
> acf(res.lm,120,lwd=2,col="chocolate1")
```





Disegnamo assieme le due serie:

```
> res.lm = ts(res.lm,frequency=4,start=c(1987,4))
> ts.plot(res.SET[6:121],res.lm,col=c("salmon1","coral1"))
```

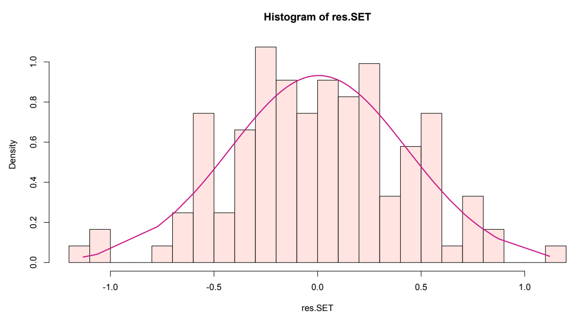


Calcoliamo le rispettive varianze spiegate dopo esserci chiariti le idee sulla precisa forma dei residui:

```
> 1 - var(res.SET)/var(tseries[3:123])
[1] 0.8143296
> 1 - var(res.lm)/var(tseries[8:123])
[1] 0.8747422
```

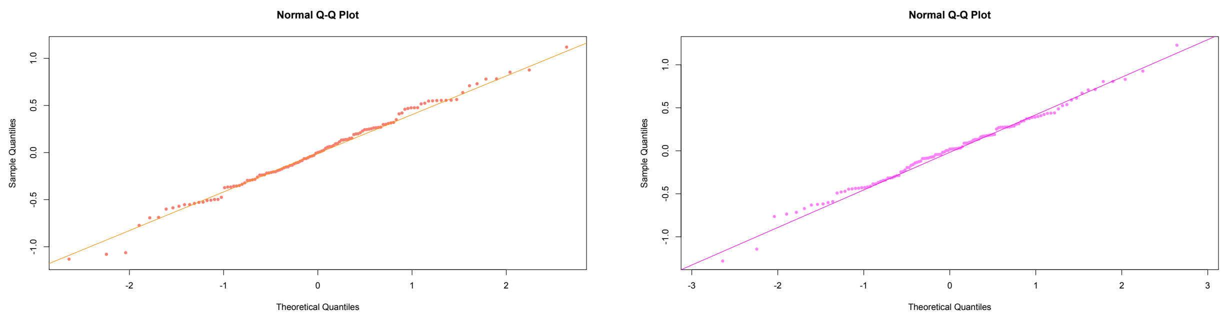
Poi rispettive medie ed istogrammi:

```
> res.SET.s = sort(res.SET)
> Z_1 = dnorm(res.SET.s,mean(res.SET),sd(res.SET))
> mean(res.SET)
[1] 0.004772491
> res.lm.s = sort(res.lm)
> Z_2 = dnorm(res.lm.s,mean(res.lm),sd(res.lm))
> mean(res.lm)
[1] 0.02610769
> hist(res.SET,20,freq=FALSE,col="mistyrose1")
> lines(res.SET.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.lm,20,freq=FALSE,col="mistyrose2")
> lines(res.lm.s,Z_2,lwd=2,col="mediumvioletred")
```



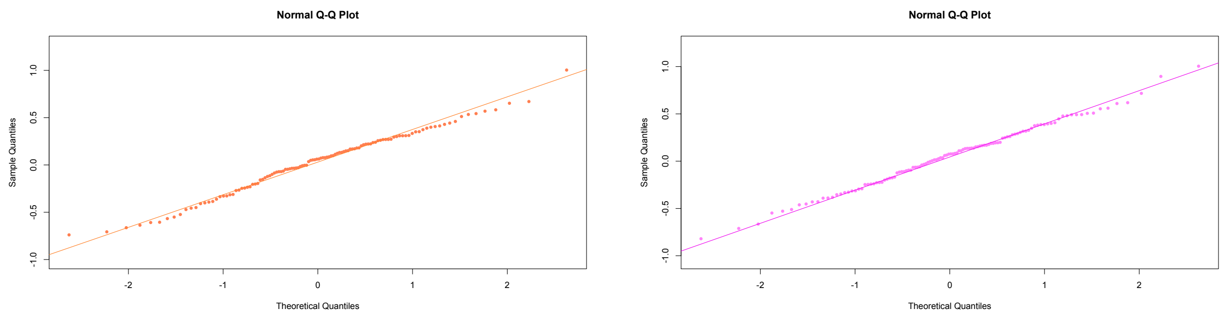
Infine i rispettivi quantili: per la prima,

```
> qqnorm(res.SET,pch=20,col="salmon",asp=1)
> qqline(res.SET,col="darkorange")
> Z_1 = rnorm(length(res.SET),mean(res.SET),sd(res.SET))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```



Per la seconda,

```
> qqnorm(res.lm,pch=20,col="coral",asp=1)
> qqline(res.lm,col="chocolate1")
> Z_2 = rnorm(length(res.lm),mean(res.lm),sd(res.lm))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```



Visto tutto, insistiamo senza dubbio alcuno nel preferire un'auto-regressione lineare rispetto ad un SET per la serie in questione, e tuttavia veniamo ora ad affrontare il finale portandocele dietro entrambe.

1.3.2 Incertezza delle previsioni

Anzitutto, due quantili per entrambe: per la prima,

```
> quantile(res.SET,0.05)
        5%
-0.6005353
> quantile(res.SET,0.95)
      95%
0.7092618
> qnorm(0.05,mean(res.SET),sd(res.SET))
[1] -0.6991341
> qnorm(0.95,mean(res.SET),sd(res.SET))
[1] 0.7086791
```

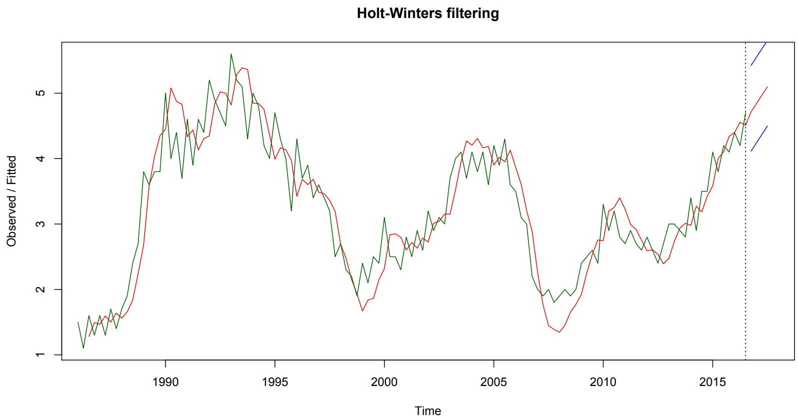
Per la seconda,

```
> quantile(res.lm,0.05)
        5%
-0.5768311
> quantile(res.lm,0.95)
      95%
0.5360492
```

```
> qnorm(0.05,mean(res.lm),sd(res.lm))
[1] -0.5228367
> qnorm(0.95,mean(res.lm),sd(res.lm))
[1] 0.5750521
```

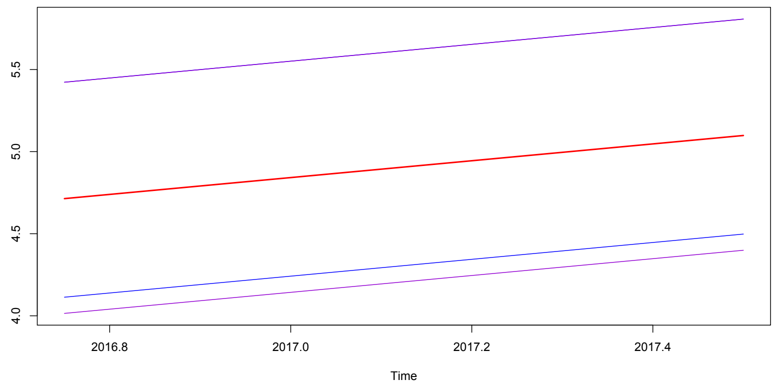
Quindi una previsione con banda di confidenza non parametrica, ossia empirica, per la prima:

```
> plot(tseries.SET,predict(tseries.SET,4),col="darkgreen")
> lines(predict(tseries.SET,4) + quantile(res.SET,0.05),col="blue")
> lines(predict(tseries.SET,4) + quantile(res.SET,0.95),col="blue")
```



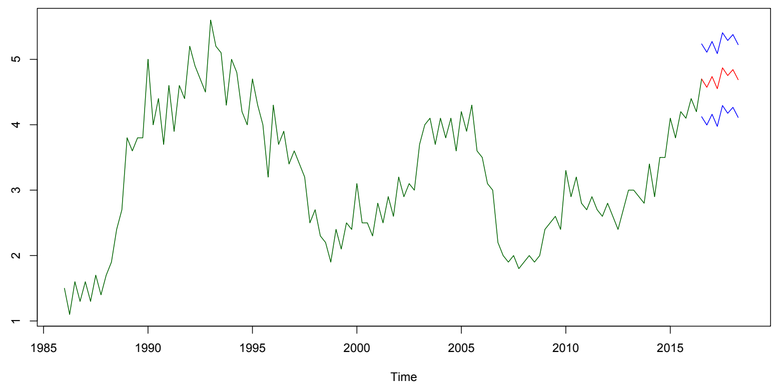
Poi un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica, ossia teorica:

```
> ts.plot(predict(tseries.SET,4),
+ predict(tseries.SET,4) + quantile(res.SET,0.05),
+ predict(tseries.SET,4) + quantile(res.SET,0.95),
+ predict(tseries.SET,4) + qnorm(0.05,mean(res.SET),sd(res.SET)),
+ predict(tseries.SET,4) + qnorm(0.95,mean(res.SET),sd(res.SET)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



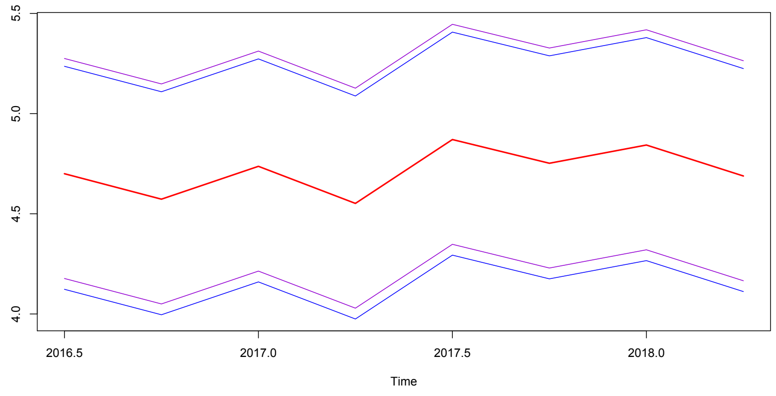
Adesso una previsione con banda di confidenza non parametrica per la seconda:

```
> ts.plot(tseries.P,pred.P,col=c("darkgreen","red"))
> lines(pred.P + quantile(res.lm,0.05),col="blue")
> lines(pred.P + quantile(res.lm,0.95),col="blue")
```



Infine un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica:

```
> pred.P.w = window(pred.P,c(2016,3))
> ts.plot(pred.P.w,
+ pred.P.w + quantile(res.lm,0.05),
+ pred.P.w + quantile(res.lm,0.95),
+ pred.P.w + qnorm(0.05,mean(res.lm),sd(res.lm)),
+ pred.P.w + qnorm(0.95,mean(res.lm),sd(res.lm)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



In conclusione, per questa serie promuoviamo con convinzione un modello basato sull'auto-regressione lineare, per altro buono a tal punto da non indurci minimamente a pensare alla necessità di migliorarlo ad auto-regressione con *fattori esogeni*, e vogliamo inoltre ripetere che la serie stessa pare di conseguenza mostrare una tendenza del trend ad un'ascesa sì piuttosto pronunciata ma soprattutto temporanea e sottomessa ad una certa ciclicità di fondo nella quale pensiamo sia il caso credere.

2 La seconda serie storica: prezzi delle case

Delle serie relative ai prezzi delle case - tutte ufficialmente disponibili ***qui*** - decidiamo di studiare la seguente, sempre intendendola come quella canonicamente associata alla tabella che proponiamo subito dopo:

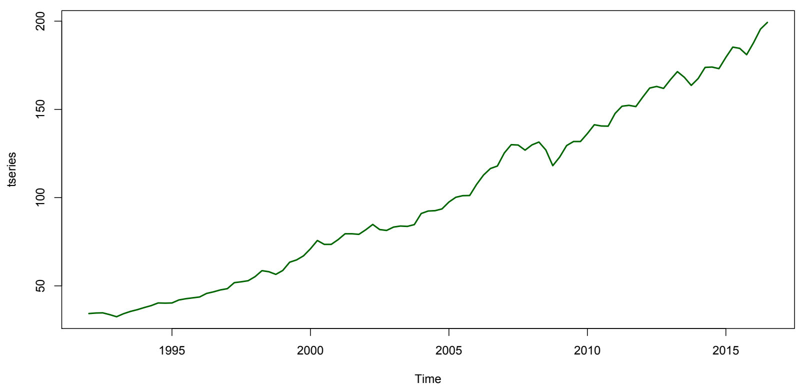
34.3 34.6 34.7 33.7 32.5 34.2 35.5 36.5 37.7 38.8 40.3 40.2 40.3 42.0 42.7 43.2 43.7 45.7 46.6 47.7 48.4 51.8 52.3 52.9 55.2 58.6 58 56.5 58.7 63.4 64.7 67 71 75.7 73.5 73.5 76.2 79.5 79.5 79.2 81.8 84.8 81.9 81.4 83.3 83.9 83.7 84.7 91 92.4 92.6 93.6 97.5 100.2 101.1 101.2 107.5 112.8 116.5 117.9 125.3 130 129.8 126.9 129.9 131.5 127 118.1 123 129.5 131.8 131.8 136.3 141.3 140.6 140.5 147.7 151.8 152.3 151.6 157 162.1 163 161.9 166.9 171.4 168.2 163.6 167.5 173.8 174 173.1 179.5 185.3 184.6 181 187.8 195.5 199.3 .

Anno	Trim. 1	Trim. 2	Trim. 3	Trim. 4
1992	34.3	34.6	34.7	33.7
1993	32.5	34.2	35.5	36.5
1994	37.7	38.8	40.3	40.2
1995	40.3	42.0	42.7	43.2
1996	43.7	45.7	46.6	47.7
1997	48.4	51.8	52.3	52.9
1998	55.2	58.6	58	56.5
1999	58.7	63.4	64.7	67
2000	71	75.7	73.5	73.5
2001	76.2	79.5	79.5	79.2
2002	81.8	84.8	81.9	81.4
2003	83.3	83.9	83.7	84.7
2004	91	92.4	92.6	93.6
2005	97.5	100.2	101.1	101.2
2006	107.5	112.8	116.5	117.9
2007	125.3	130	129.8	126.9
2008	129.9	131.5	127	118.1
2009	123	129.5	131.8	131.8
2010	136.3	141.3	140.6	140.5
2011	147.7	151.8	152.3	151.6
2012	157	162.1	163	161.9
2013	166.9	171.4	168.2	163.6
2014	167.5	173.8	174	173.1
2015	179.5	185.3	184.6	181
2016	187.8	195.5	199.3	NA

Così, di nuovo, da una parte abbiamo una serie già pronta per esser agilmente caricata su **R**, mentre d'altra parte abbiamo pure la sua forma più completa e chiara della quale conviene restar sempre ben consapevoli - già a cominciare dalla scelta dei due parametri di partenza `frequency` e `start` del codice.

Evidenziamo quindi tale serie, copiamola mediante ad esempio l'usuale combinazione di tasti `cmd+c`, e procediamo su **R** coi primi comandi ed il primo grafico a riguardo:

```
> X <- scan(pipe("pbpaste"))
Read 99 items
> tseries = ts(X,frequency=4,start=c(1992,1))
> ts.plot(tseries,lwd=2,col="darkgreen")
```



A primo impatto la serie così rappresentata sembrerebbe delle più semplici, in quanto parrebbe affetta da rumore solo in misura davvero contenuta ed in quanto parrebbe presentare per di più un trend ascendente alquanto palese ed al contrario una stagionalità nient'affatto evidente, per cui in particolare non saremmo affatto tentati di ripensare alla nostra scelta iniziale del paramentro `frequency`.

Certo è che andrebbe osservato come per una tale tipologia di dati potrebbe invece aver benissimo senso una reale periodicità, seppur di entità eventualmente davvero modesta.

Detto questo, resta comunque opportuna una vera analisi matematica tecnica a riguardo.

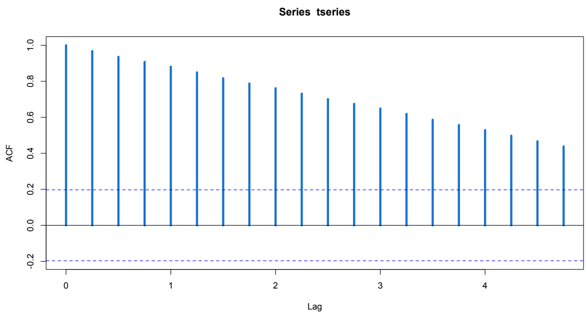
2.1 Funzione di auto-correlazione, decomposizioni della serie e analisi dei residui

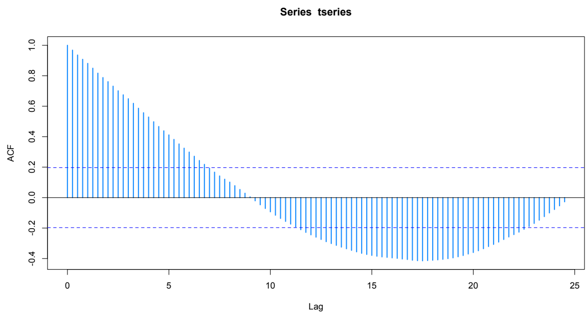
Agiamo sulla serie per intero riprendendo i comandi digitati poco fa.

2.1.1 Funzione di auto-correlazione

Partiamo di nuovo con la ACF su due lag estremali:

```
> acf(tseries,lwd=4,col="dodgerblue3")
> acf(tseries,100,lwd=2,col="dodgerblue")
```



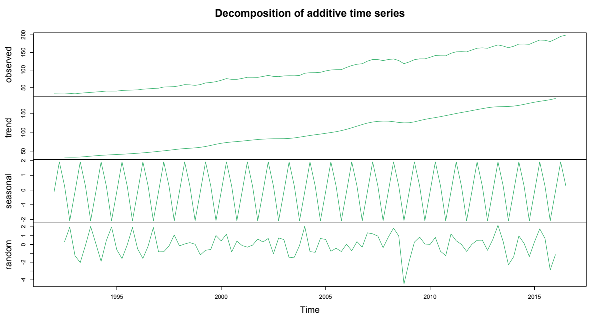


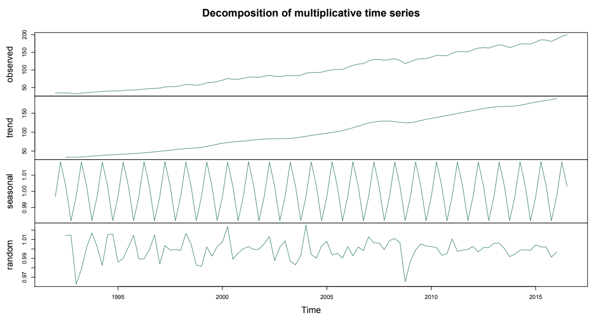
Completamente secondo le nostre impressioni visive, ed in particolare teniamo il `frequency=4`.

2.1.2 Decomposizioni della serie

Decomposizione standard additiva e decomposizione moltiplicativa rispettivamente:

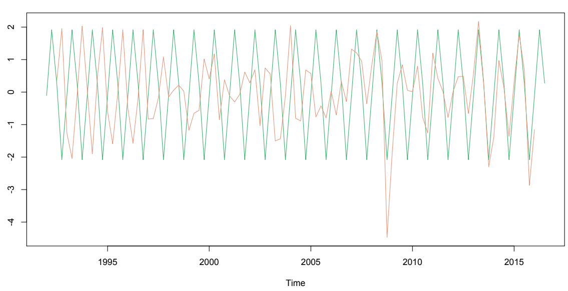
```
> dec.add = decompose(tseries)
> dec.mult = decompose(tseries,type="multiplicative")
> plot(dec.add,col="mediumseagreen")
> plot(dec.mult,col="aquamarine4")
```

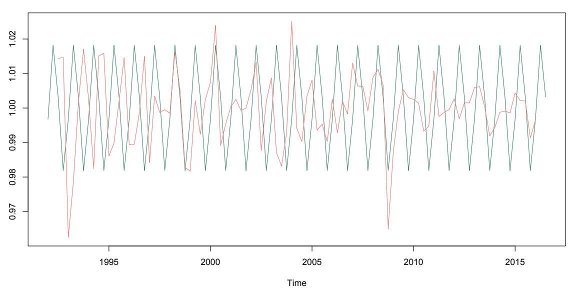




Così, in effetti, entrambe le decomposizioni rivelano una stagionalità ed un rumore di un medesimo ordine di grandezza che risulta del tutto trascurabile rispetto a quello del trend. Vediamolo più chiaramente: da una parte,

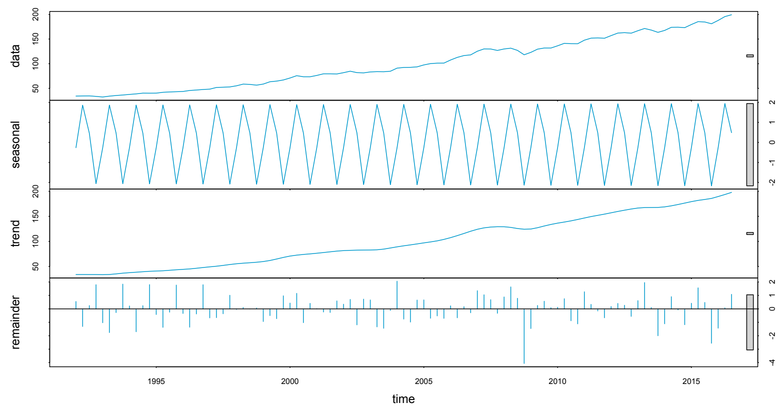
```
> ts.plot(dec.add$seasonal,dec.add$random,col=c("mediumseagreen","darksalmon"))
> ts.plot(dec.mult$seasonal,dec.mult$random,col=c("aquamarine4","lightcoral"))
```





D'altra parte, soprattutto,

```
> plot(stl(tseries,100),col="deepskyblue3")
```



In definitiva, la struttura della serie risulta veramente di trend assolutamente dominante: cerchiamo adesso di capire quale delle due decomposizioni la descrivi meglio, senza indagare invece quanto offrirebbero degli `stl` più oculati.

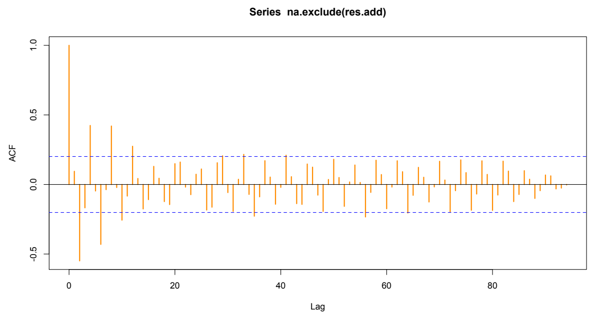
2.1.3 Analisi dei residui

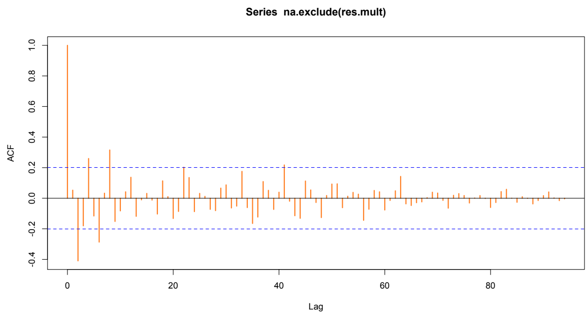
Rendiamo omogenee le due serie dei residui e confrontiamole poi al meglio delle nostre possibilità:

```
> res.add = dec.add$random
> res.mult = dec.mult$random
> trend.mult = dec.mult$trend
> res.mult = mean(na.exclude(trend.mult)) * (res.mult - 1)
```

Rispettive funzioni di auto-correlazione:

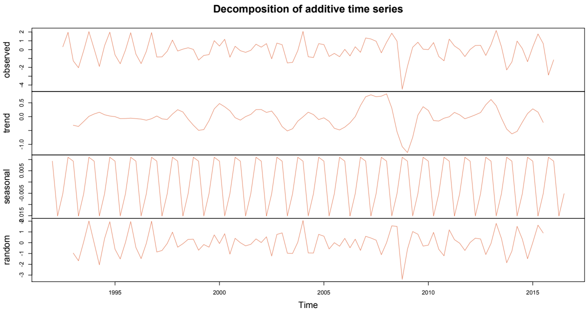
```
> acf(na.exclude(res.add),100,lwd=2,col="darkorange")
> acf(na.exclude(res.mult),100,lwd=2,col="chocolate1")
```

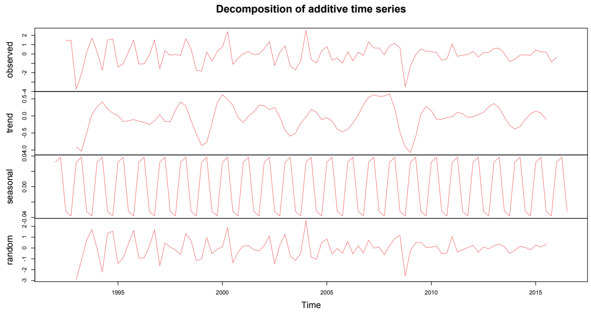




Rispettive decomposizioni additive:

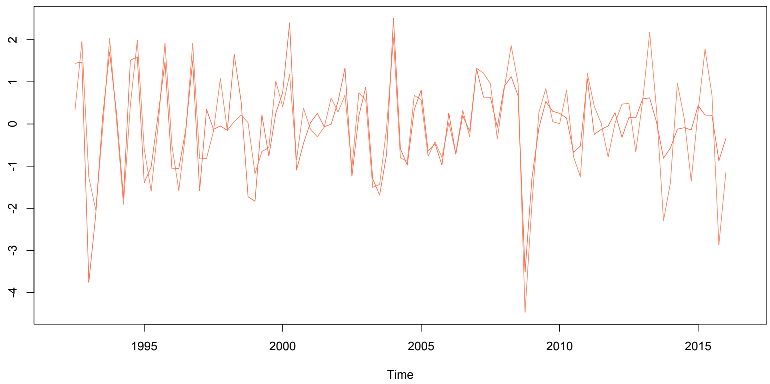
```
> plot(decompose(res.add),col="darksalmon")
> plot(decompose(res.mult),col="lightcoral")
```





Le due serie semplicemente disegnate assieme:

```
> ts.plot(res.add,res.mult,col=c("salmon1","coral1"))
```



Rispettive varianze spiegate calcolate tenendo conto dei rispettivi dati `NA` presenti:

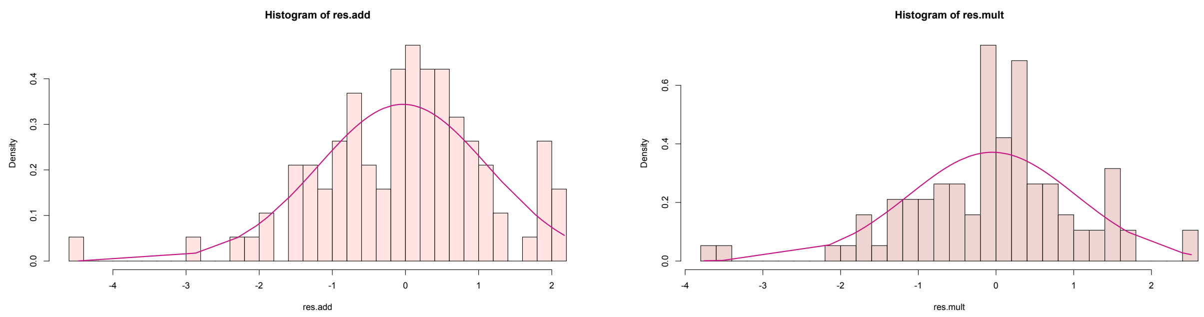
```
> 1 - var(na.exclude(res.add))/var(tseries[3:97])
[1] 0.9994008
> 1 - var(na.exclude(res.mult))/var(tseries[3:97])
[1] 0.9994859
```

Solita curiosità a riguardo della seconda serie dei residui:

```
> 1 - var(na.exclude(log(dec.mult$random)))/var(tseries[3:97])
[1] 0.9999999
```

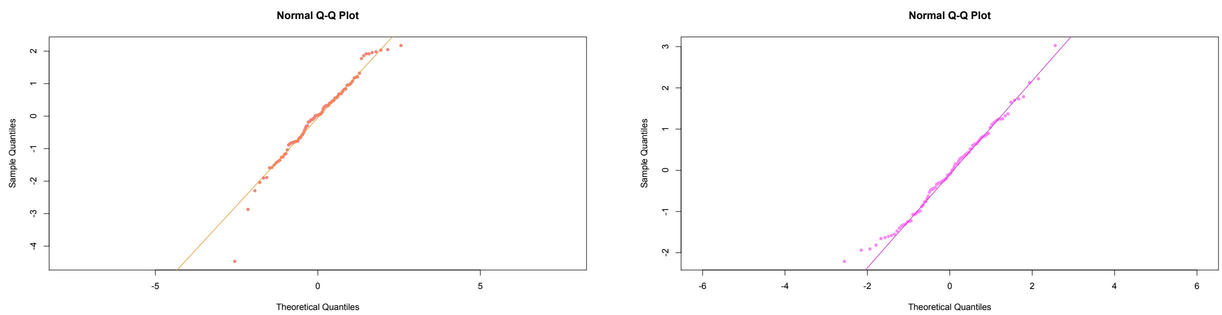
Rispettive medie ed istogrammi:

```
> res.add.s = sort(na.exclude(res.add))
> Z_1 = dnorm(res.add.s,mean(res.add.s),sd(res.add.s))
> mean(res.add.s)
[1] -0.03470335
> res.mult.s = sort(na.exclude(res.mult))
> Z_2 = dnorm(res.mult.s,mean(res.mult.s),sd(res.mult.s))
> mean(res.mult.s)
[1] -0.0468046
> hist(res.add,25,freq=FALSE,col="mistyrose1")
> lines(res.add.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.mult,25,freq=FALSE,col="mistyrose2")
> lines(res.mult.s,Z_2,lwd=2,col="mediumvioletred")
```



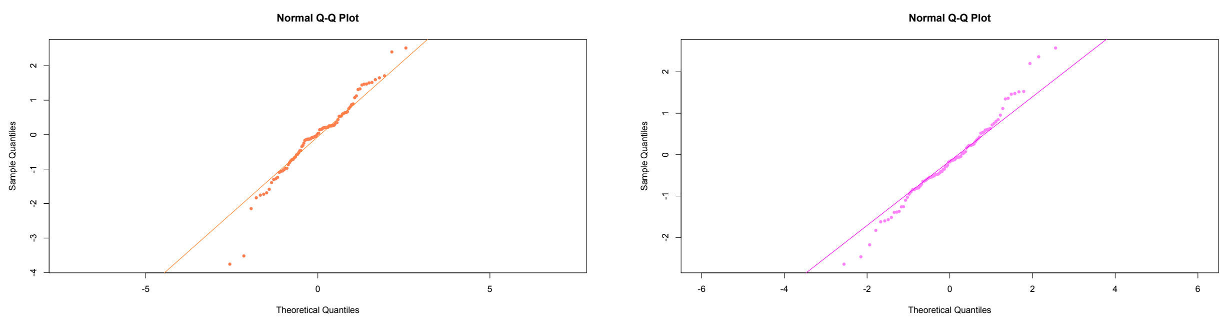
Rispettivi quantili, finalmente: per la prima,

```
> qqnorm(res.add.s,pch=20,col="salmon",asp=1)
> qqline(res.add.s,col="darkorange")
> Z_1 = rnorm(length(res.add.s),mean(res.add.s),sd(res.add.s))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```



Per la seconda,

```
> qqnorm(res.mult.s,pch=20,col="coral",asp=1)
> qqline(res.mult.s,col="chocolate1")
> Z_2 = rnorm(length(res.mult.s),mean(res.mult.s),sd(res.mult.s))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```



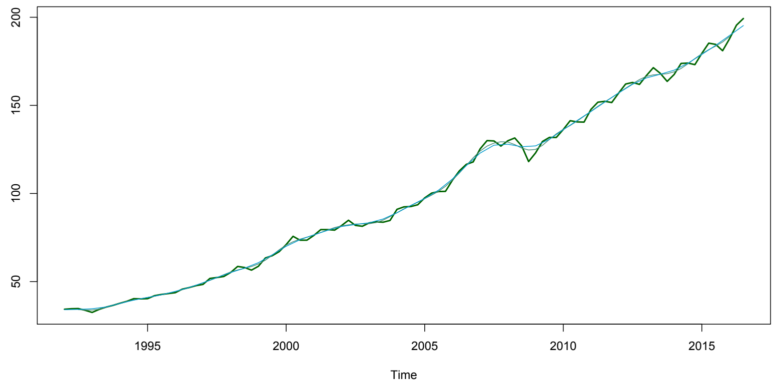
Visto tutto, concludiamo che le due decomposizioni sono parecchio simili e ben riuscite, e dell'analisi dei relativi residui apprezziamo specialmente le varianze spiegate ammettendo d'altro canto che purtroppo i due `qqnorm` potevano esser ben più soddisfacenti, riferendosi specialmente a quello della seconda serie di residui. Dopo di ché, in verità, saremmo comunque tentati di considerare come migliore la decomposizione moltiplicativa, a maggior ragione se pensiamo al fatto che i dati della serie in analisi sono poco numerosi; detto questo, forse la scelta di quella additiva lascerebbe con meno dubbi preoccupanti.

2.2 Analisi e previsioni del trend e della serie

Ancora consideriamo opportuna una previsione dell'andamento della serie in un prossimo futuro - nell'anno 2017 - la quale sia il risultato del confronto di analisi e previsione del solo trend con analisi e previsione di un filtraggio della serie il quale sia una specie di via di mezzo proprio tra trend e serie stessa.

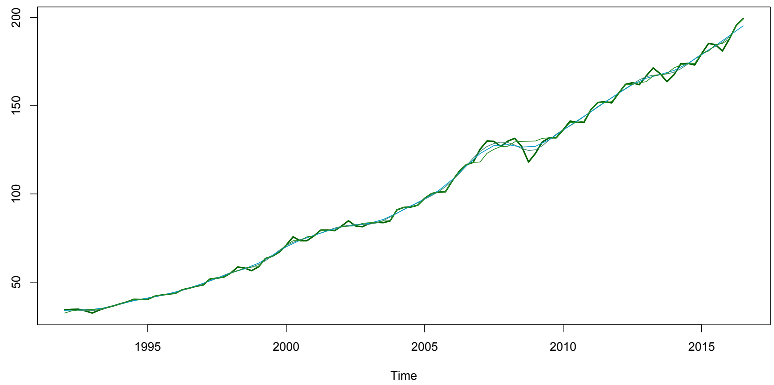
Cominciamo con calma tracciando insieme la serie, il suo trend ed anche uno smoothing del medesimo per poi analizzarli e quindi prevederli separatamente uno ad uno, convenendo naturalmente di esser ben più accurati nel farlo per l'intera serie:

```
> ts.plot(tseries,trend.mult,stl(tseries,4)$time.series[,2],lwd=c(2,1,1),
+ col=c("darkgreen","aquamarine4","deepskyblue3"))
```



Per curiosità, aggiungiamo al grafico il `sort` della serie:

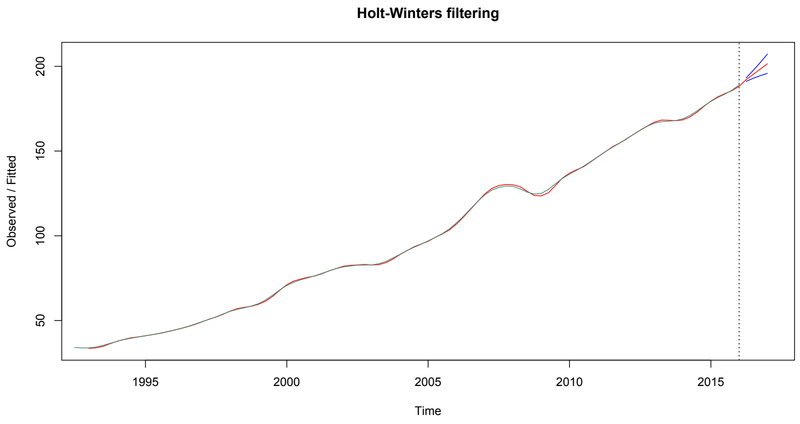
```
> ts.plot(tseries,trend.mult,stl(tseries,4)$time.series[,2],sort(tseries),
+ lwd=c(2,1,1,1),col=c("darkgreen","aquamarine4","deepskyblue3","forestgreen"))
```



2.2.1 Analisi e previsioni del trend

Partiamo da un SET automatico per il trend:

```
> trend = trend.mult
> trend.SET = HoltWinters(na.omit(trend),gamma=FALSE)
> plot(trend.SET,predict(trend.SET,4,prediction.interval=TRUE),col="aquamarine4")
```

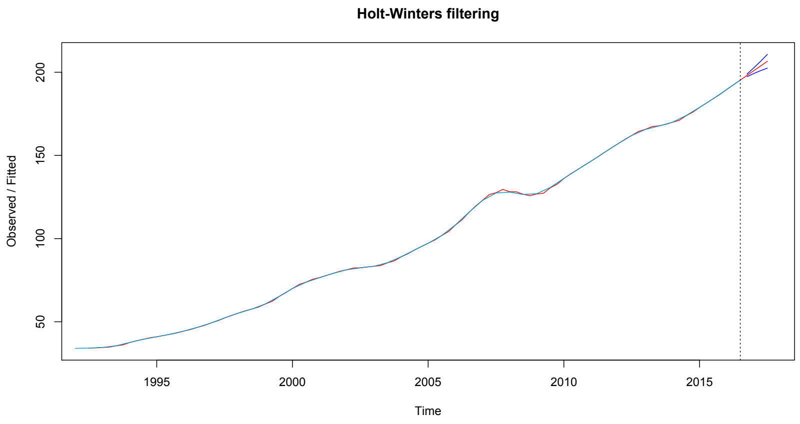


Una nota:

```
> trend.SET

Smoothing parameters:
 alpha: 1
 beta : 1
 gamma: FALSE
```

Guardiamo quindi il SET automatico per lo smoothing:

```
> trend = stl(tseries,4)$time.series[,2]
> trend.SET = HoltWinters(trend,gamma=FALSE)
> plot(trend.SET,predict(trend.SET,4,prediction.interval=TRUE),col="deepskyblue3")
```

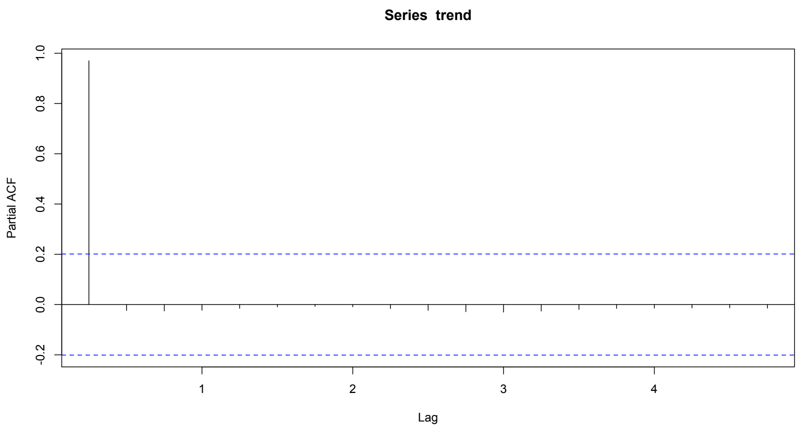


Una nota:

```
> trend.SET

Smoothing parameters:
 alpha: 0.9291452
 beta : 1
 gamma: FALSE
```

Procediamo adesso con un'auto-regressione lineare effettuata manualmente per il trend, basandosi sulla funzione di auto-correlazione parziale per il trend stesso:

```
> trend = na.omit(trend.mult)
> pacf(trend)
```



Costruiamo un modello lineare per il trend:

```
> n = length(trend)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(trend.lm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.15465    0.07497   2.063  0.04236 *
A[, 1]       0.35202    0.10678   3.297  0.00146 **
A[, 2]      -1.04609    0.34161  -3.062  0.00299 **
A[, 3]       1.38074    0.52659   2.622  0.01046 *
A[, 4]      -1.59141    0.57990  -2.744  0.00748 **
A[, 5]       2.71950    0.52827   5.148 1.84e-06 ***
A[, 6]      -3.88435    0.34085 -11.396  < 2e-16 ***
A[, 7]       3.07114    0.10591  28.998  < 2e-16 ***

Residual standard error: 0.2502 on 80 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 4.084e+05 on 7 and 80 DF,  p-value: < 2.2e-16
```

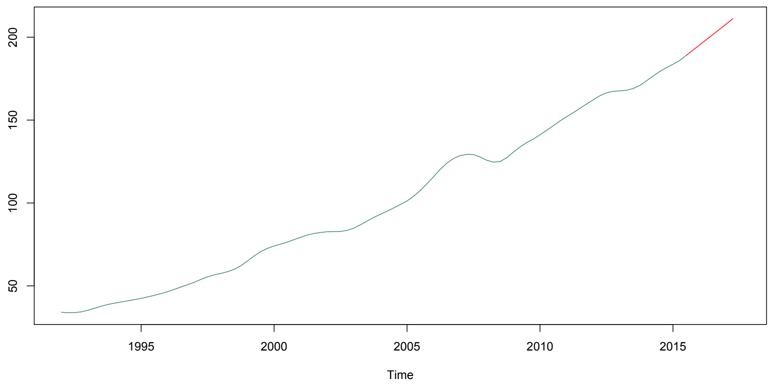
Ottimizziamo tale modello:

```
> trend.lm = lm(A[,8]~A[,6]+A[,7]+0)
> summary(trend.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 6] -0.90674    0.04892  -18.53   <2e-16 ***
A[, 7]  1.90828    0.04819   39.60   <2e-16 ***

Residual standard error: 0.532 on 86 degrees of freedom
```

```
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 2.044e+06 on 2 and 86 DF,  p-value: < 2.2e-16
```

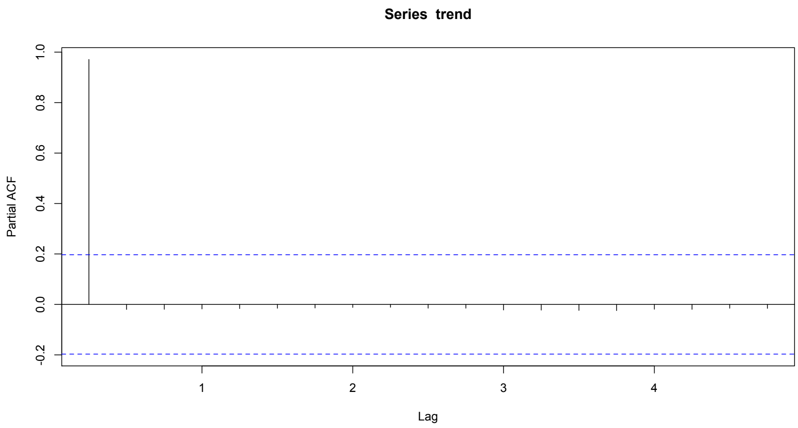
Conseguente previsione per il trend:

```
> pred = 1:(n-k+1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> trend.P = ts(trend.P,frequency=4,start=c(1992,1))
> pred.P = ts(pred.P,frequency=4,start=c(1992,1))
> ts.plot(trend.P,pred.P,col=c("aquamarine4","red"))
```



Volendo ora far lo stesso per lo smoothing, guardiamo la sua funzione di auto-correlazione parziale:

```
> trend = stl(tseries,4)$time.series[,2]
> pacf(trend)
```



Costruiamo un modello lineare per lo smoothing:

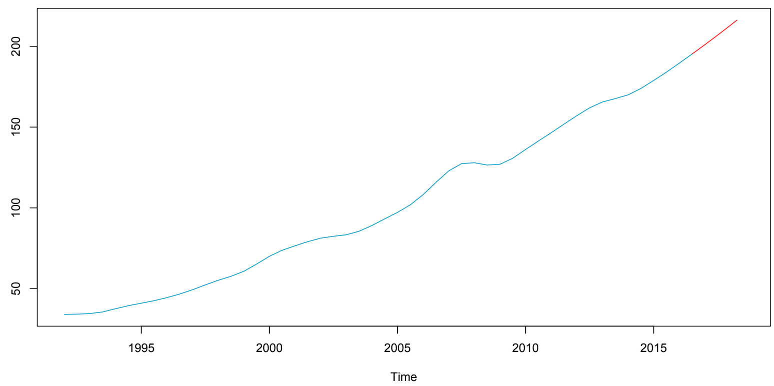
```
> n = length(trend)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(trend.lm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12276    0.06584   1.865  0.06572 .
A[, 1]       0.07459    0.10874   0.686  0.49462
A[, 2]      -0.77633    0.21916  -3.542  0.00065 ***
A[, 3]       1.39446    0.15784   8.835 1.28e-13 ***
A[, 4]       0.24857    0.26019   0.955  0.34214
A[, 5]      -1.91129    0.15785 -12.108  < 2e-16 ***
A[, 6]       0.05684    0.21945   0.259  0.79627
A[, 7]       1.91409    0.10869  17.610  < 2e-16 ***

Residual standard error: 0.2225 on 84 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 5.889e+05 on 7 and 84 DF,  p-value: < 2.2e-16
```

Ottimizziamo tale modello:

```
> trend.lm = lm(A[,8]~A[,6]+A[,7]+0)
> summary(trend.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 6] -0.90315    0.04890  -18.47   <2e-16 ***
A[, 7]  1.90476    0.04817   39.55   <2e-16 ***

Residual standard error: 0.4247 on 90 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 3.426e+06 on 2 and 90 DF,  p-value: < 2.2e-16
```

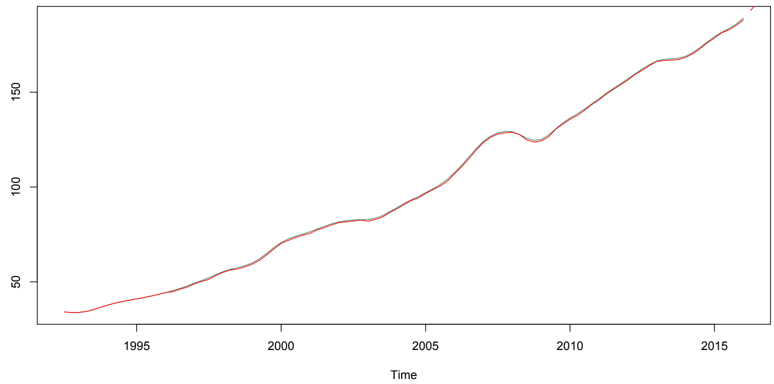
Conseguente previsione per lo smoothing:

```
> pred = 1:(n-k+1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> trend.P = ts(trend.P,frequency=4,start=c(1992,1))
> pred.P = ts(pred.P,frequency=4,start=c(1992,1))
> ts.plot(trend.P,pred.P,col=c("deepskyblue3","red"))
```



Affianchiamo al tutto pure un metodo automatico dei minimi quadrati nell'auto-regressione lineare per il trend, e solo per questo:

```
> trend = na.omit(trend.mult)
> trend.ar = ar(trend,method="ols",aic=TRUE)

> trend.ar

Order selected 15  sigma^2 estimated as  0.03935

> 1 - trend.ar$var/var(trend)
          [,1]
[1,] 0.9999825

> o = trend.ar$order
> a = trend.ar$ar
> trend.ar.P = trend
> for(i in (o+1):length(trend)){
+ trend.ar.P[i] = sum(rev(a)*trend[(i-o):(i-1)]) + mean(trend)*(1-sum(a))
+ }
> ts.plot(trend,trend.ar.P,col=c("aquamarine4","red"))
> lines(predict(trend.ar,n.ahead=8,se.fit=FALSE),col="red")
```

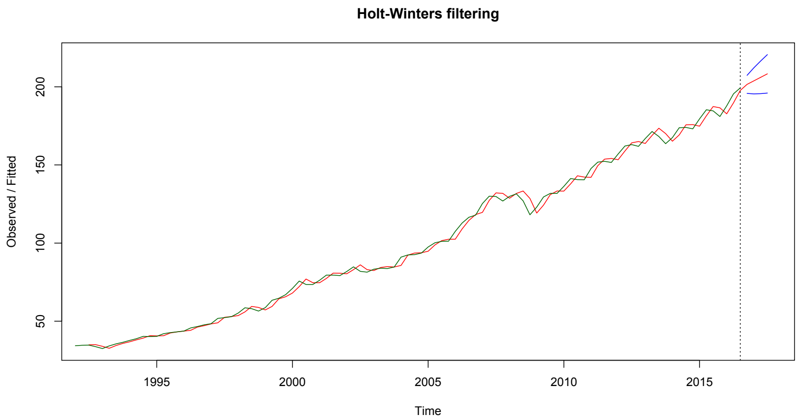


Siamo dunque alquanto soddisfatti specie dell'auto-regressione lineare svolta per il trend e per lo smoothing scelto, in quanto conferma del tutto una netta tendenza del trend ad un'ascesa in assoluto accordo con quanto chiunque si aspetterebbe, e di conseguenza ci sentiamo più che fiduciosi in un'auto-regressione lineare anche per l'analisi e la previsione dell'intera serie contando poi in conclusioni analoghe.

2.2.2 Analisi e previsioni della serie

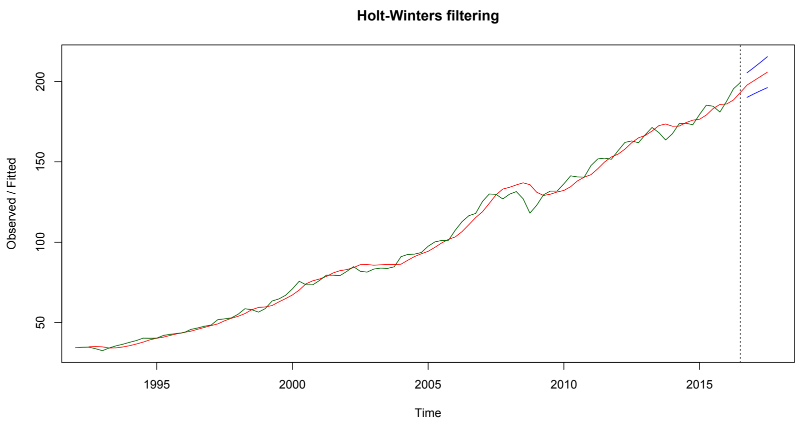
Per cominciare, proponiamo un SET automatico:

```
> tseries.SET = HoltWinters(tseries,gamma=FALSE)
> plot(tseries.SET,predict(tseries.SET,4,prediction.interval=TRUE),col="darkgreen")
> tseries.SET

Smoothing parameters:
 alpha: 1
 beta : 0.04363526
 gamma: FALSE
```



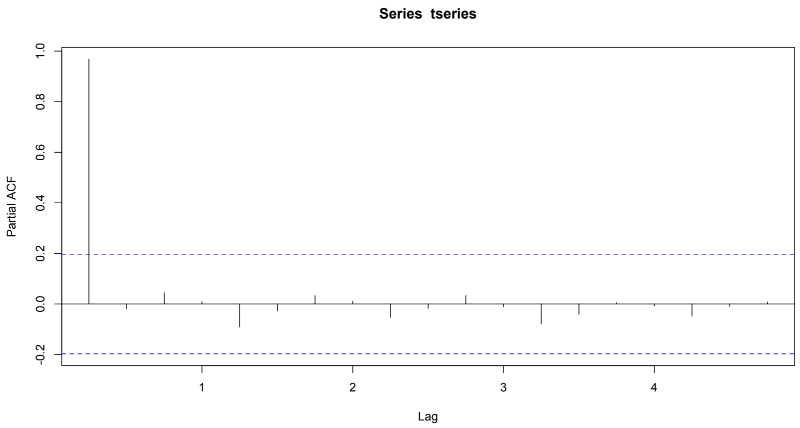
Optiamo però per il seguente SET calibrato manualmente, basandoci com'è naturale su diverse e ragionate sperimentazioni condotte a riguardo:

```
> tseries.SET = HoltWinters(tseries,alpha=0.325,beta=0.175,gamma=FALSE)
> plot(tseries.SET,predict(tseries.SET,4,prediction.interval=TRUE),col="darkgreen")
```



Andiamo ora di auto-regressione lineare manuale partendo dalla funzione di auto-correlazione parziale:

```
> pacf(tseries)
```



Costruiamo un modello lineare:

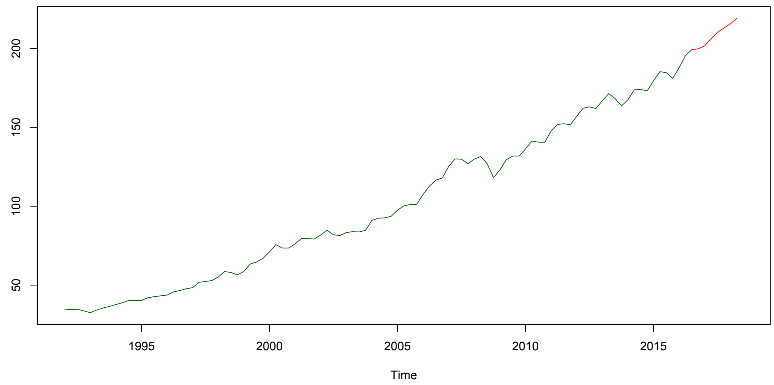
```
> n = length(tseries)
> k = 8
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = tseries[j:(n-k+j)]
+ }
> tseries.lm = lm(A[,8]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7])
> summary(tseries.lm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8010     0.4758   1.683 0.096019 .
A[, 1]        0.1561     0.1148   1.360 0.177479
A[, 2]        0.3273     0.1985   1.649 0.102852
A[, 3]       -1.0436     0.1778  -5.870 8.42e-08 ***
A[, 4]        0.6832     0.1618   4.222 6.10e-05 ***
A[, 5]        0.1137     0.1722   0.660 0.510859
A[, 6]       -0.6560     0.1896  -3.460 0.000852 ***
A[, 7]        1.4286     0.1101  12.982  < 2e-16 ***

Residual standard error: 1.788 on 84 degrees of freedom
Multiple R-squared:  0.9987,Adjusted R-squared:  0.9986
F-statistic:  9161 on 7 and 84 DF,  p-value: < 2.2e-16
```

Ottimizziamo tale modello:

```
> tseries.lm = lm(A[,8]~A[,5]+A[,6]+A[,7]+0)
> summary(tseries.lm)

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
A[, 5]  0.60094    0.08854   6.787 1.23e-09 ***
A[, 6] -0.84605    0.13280  -6.371 8.07e-09 ***
A[, 7]  1.26583    0.08603  14.714  < 2e-16 ***

Residual standard error: 2.42 on 89 degrees of freedom
Multiple R-squared:  0.9996,Adjusted R-squared:  0.9996
F-statistic: 7.037e+04 on 3 and 89 DF,  p-value: < 2.2e-16
```

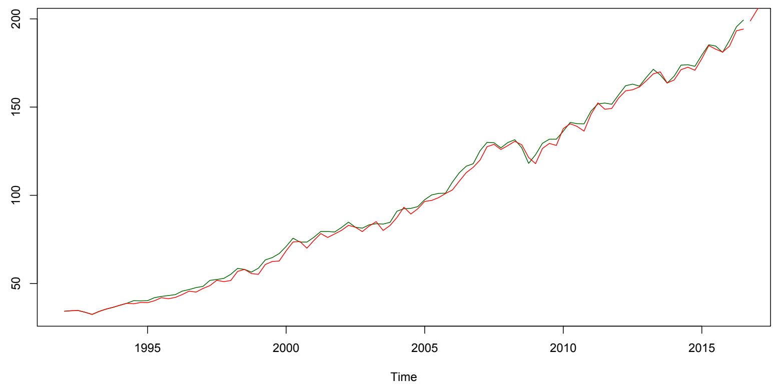
Conseguente previsione:

```
> pred = 1:(n+k-1)
> pred[1:n] = tseries
> for(i in 1:(k-1)){
+ pred[n+i] = coef(tseries.lm) %*% c(pred[n+i-3],pred[n+i-2],pred[n+i-1])
+ }
> tseries.P = pred.P = pred
> for(i in (n+1):(n+8)){
+ tseries.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> tseries.P = ts(tseries.P,frequency=4,start=c(1992,1))
> pred.P = ts(pred.P,frequency=4,start=c(1992,1))
> ts.plot(tseries.P,pred.P,col=c("darkgreen","red"))
```



Occhiata al metodo automatico dei minimi quadrati nell'auto-regressione lineare:

```
> tseries.ar = ar(tseries,method="ols",aic=TRUE)

> tseries.ar

Order selected 10  sigma^2 estimated as  2.54

> 1 - tseries.ar$var/var(tseries)
          [,1]
[1,] 0.9989569

> o = tseries.ar$order
> a = tseries.ar$ar
> tseries.ar.P = tseries
> for(i in (o+1):length(tseries)){
+ tseries.ar.P[i] = sum(rev(a)*tseries[(i-o):(i-1)]) + mean(tseries)*(1-sum(a))
+ }
> ts.plot(tseries,tseries.ar.P,col=c("darkgreen","red"))
> lines(predict(tseries.ar,n.ahead=8,se.fit=FALSE),col="red")
```



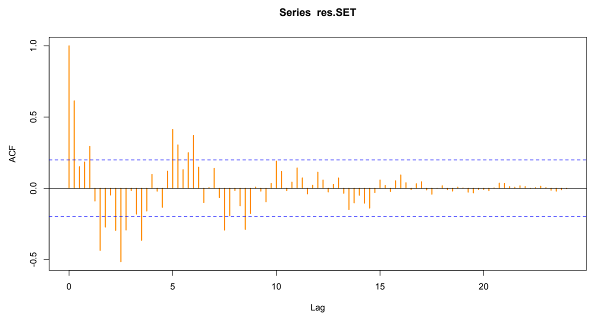
Così non abbiamo avuto assolutamente alcuna sorpresa: buttiamoci subito sui residui.

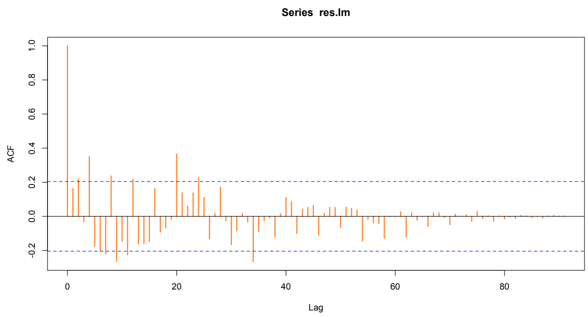
2.3 Analisi dei residui ed incertezza delle previsioni

2.3.1 Analisi dei residui

Estraiamo le due serie dei residui ed osserviamo le rispettive funzioni di auto-correlazione:

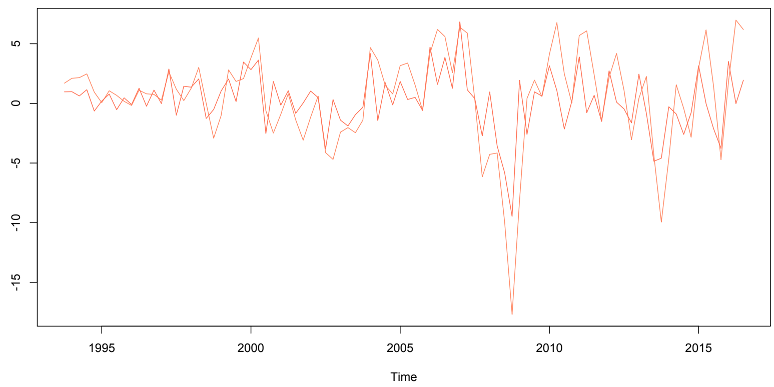
```
> res.SET = residuals(tseries.SET)
> length(res.SET)
[1] 97
> res.lm = residuals(tseries.lm)
> length(res.lm)
[1] 92
> acf(res.SET,100,lwd=2,col="darkorange")
> acf(res.lm,95,lwd=2,col="chocolate1")
```






```
> res.lm = ts(res.lm,frequency=4,start=c(1993,4))
> ts.plot(res.SET[6:97],res.lm,col=c("salmon1","coral1"))
```

Disegnamo assieme le due serie:

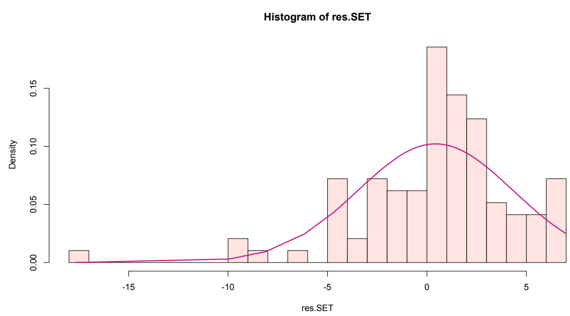


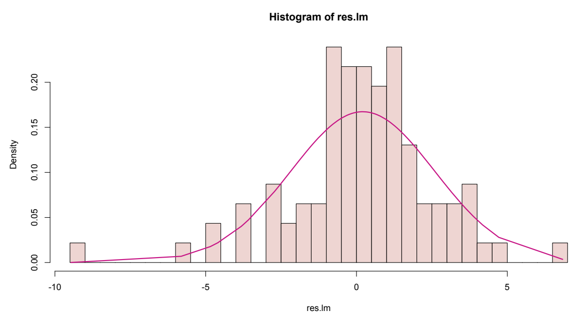
Calcoliamo le rispettive varianze spiegate dopo esserci chiariti le idee sulla precisa forma dei residui:

```
> 1 - var(res.SET)/var(tseries[3:99])
[1] 0.9936224
> 1 - var(res.lm)/var(tseries[8:99])
[1] 0.9974787
```

Poi rispettive medie ed istogrammi:

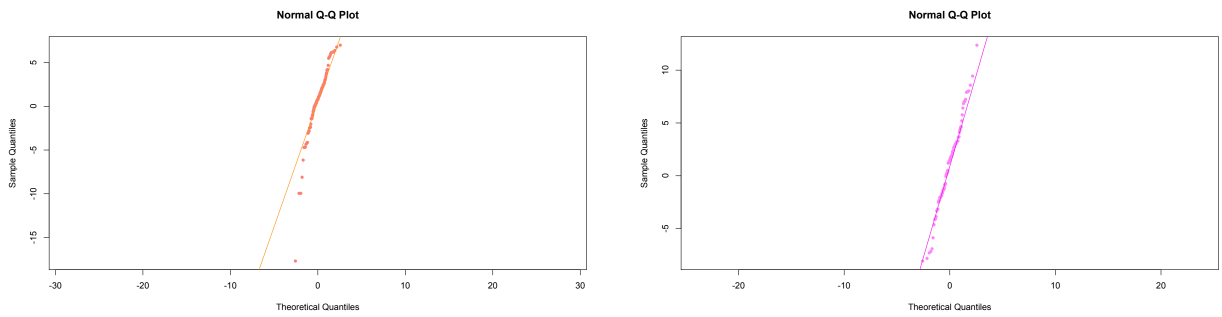
```
> res.SET.s = sort(res.SET)
> Z_1 = dnorm(res.SET.s,mean(res.SET),sd(res.SET))
> mean(res.SET)
[1] 0.4307148
> res.lm.s = sort(res.lm)
> Z_2 = dnorm(res.lm.s,mean(res.lm),sd(res.lm))
> mean(res.lm)
[1] 0.207686
> hist(res.SET,20,freq=FALSE,col="mistyrose1")
> lines(res.SET.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.lm,25,freq=FALSE,col="mistyrose2")
> lines(res.lm.s,Z_2,lwd=2,col="mediumvioletred")
```





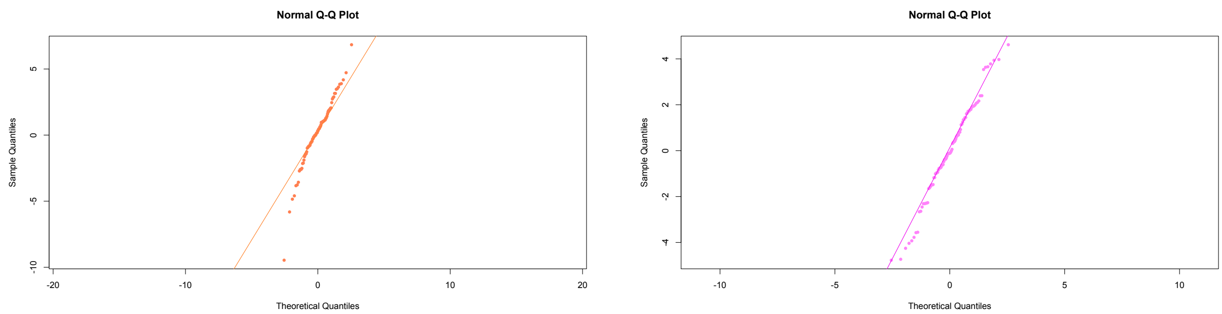
Infine i rispettivi quantili: per la prima,

```
> qqnorm(res.SET,pch=20,col="salmon",asp=1)
> qqline(res.SET,col="darkorange")
> Z_1 = rnorm(length(res.SET),mean(res.SET),sd(res.SET))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```



Per la seconda,

```
> qqnorm(res.lm,pch=20,col="coral",asp=1)
> qqline(res.lm,col="chocolate1")
> Z_2 = rnorm(length(res.lm),mean(res.lm),sd(res.lm))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```



Visto tutto, insisteremmo di certo nel preferire un'auto-regressione lineare rispetto ad un SET per la serie in questione se non fosse per l'aspetto dei rispettivi `qqnorm`; ricordiamo però che i dati della serie in analisi sono poco numerosi, e comuque sia veniamo ora ad affrontare il finale portandocele dietro entrambe.

2.3.2 Incertezza delle previsioni

Anzitutto, due quantili per entrambe: per la prima,

```
> quantile(res.SET,0.05)
       5%
-5.003715
> quantile(res.SET,0.95)
     95%
6.16597
> qnorm(0.05,mean(res.SET),sd(res.SET))
[1] -5.993405
> qnorm(0.95,mean(res.SET),sd(res.SET))
[1] 6.854834
```

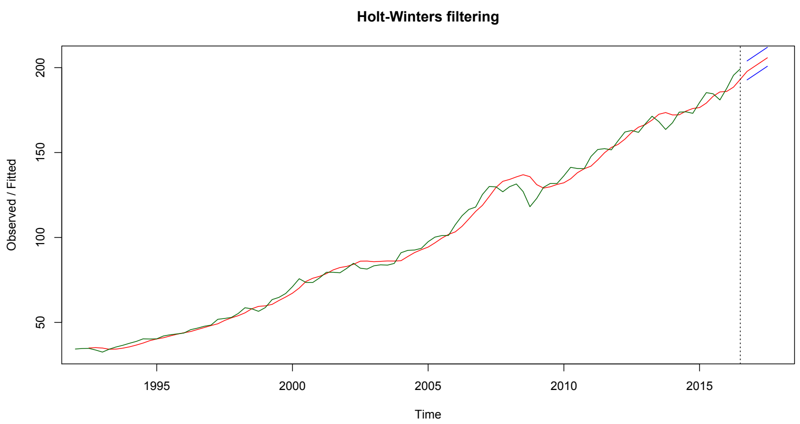
Per la seconda:

```
> quantile(res.lm,0.05)
       5%
-3.795838
> quantile(res.lm,0.95)
      95%
```

```
3.729149
> qnorm(0.05,mean(res.lm),sd(res.lm))
[1] -3.714013
> qnorm(0.95,mean(res.lm),sd(res.lm))
[1] 4.129385
```

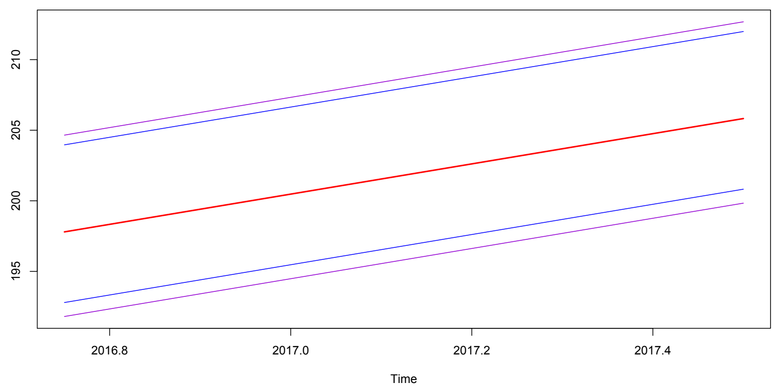
Un po' meno deludenti tali specifici quantili! Quindi, ora, una previsione con banda di confidenza non parametrica per la prima:

```
> plot(tseries.SET,predict(tseries.SET,4),col="darkgreen")
> lines(predict(tseries.SET,4) + quantile(res.SET,0.05),col="blue")
> lines(predict(tseries.SET,4) + quantile(res.SET,0.95),col="blue")
```



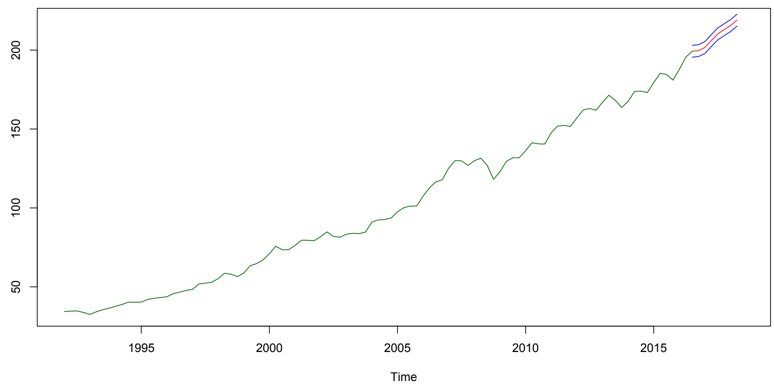
Poi un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica:

```
> ts.plot(predict(tseries.SET,4),
+ predict(tseries.SET,4) + quantile(res.SET,0.05),
+ predict(tseries.SET,4) + quantile(res.SET,0.95),
+ predict(tseries.SET,4) + qnorm(0.05,mean(res.SET),sd(res.SET)),
+ predict(tseries.SET,4) + qnorm(0.95,mean(res.SET),sd(res.SET)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



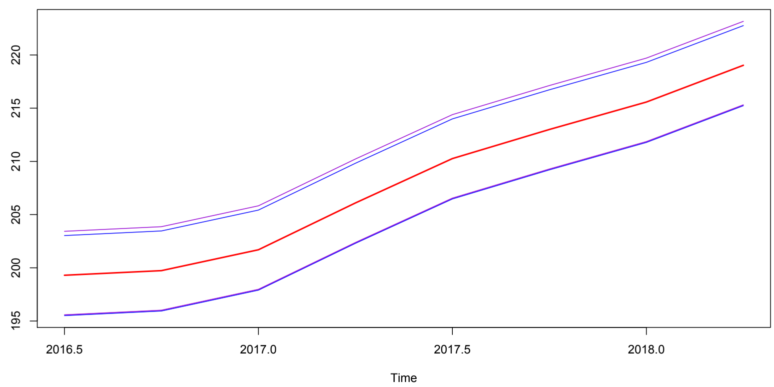
Adesso una previsione con banda di confidenza non parametrica per la seconda:

```
> ts.plot(tseries.P,pred.P,col=c("darkgreen","red"))
> lines(pred.P + quantile(res.lm,0.05),col="blue")
> lines(pred.P + quantile(res.lm,0.95),col="blue")
```



Infine un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica:

```
> pred.P.w = window(pred.P,c(2016,3))
> ts.plot(pred.P.w,
+ pred.P.w + quantile(res.lm,0.05),
+ pred.P.w + quantile(res.lm,0.95),
+ pred.P.w + qnorm(0.05,mean(res.lm),sd(res.lm)),
+ pred.P.w + qnorm(0.95,mean(res.lm),sd(res.lm)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



In conclusione, anche per questa serie promuoviamo con una certa convinzione un modello basato sull'auto-regressione lineare, per altro buono a tal punto da non indurci minimamente a pensare alla necessità di levigarlo ad auto-regressione mediante fattori esogeni, e teniamo inoltre a ripetere che la serie stessa mostra di conseguenza la sola tendenza del trend al persistere in un'ascesa piuttosto pronunciata, e questo praticamente senza alcun margine considerevole di errore.

3 La terza serie storica: spese nell'arredamento

Delle serie relative alle spese nell'arredamento di interni generici - tutte ufficialmente disponibili ***qui*** - decidiamo di studiare la seguente, intendendola di nuovo come quella canonicamente associata alla tabella che proponiamo subito dopo:

50.4 40.5 40.9 40 38.6 44.3 43.1 46.1 52.4 58.3 57.4 76.9 51.7 44.3 43.5 42.1 42.2 48.6 47.8 52 58.9 64.2 67 87.1 59.9 47.9 50.2 49.3 52.4 58.2 54.5 55.6 64 71.9 77.2 104.3 68.1 53.2 55.5 54.4 53 60.1 58.1 61.2 66.4 74.4 71.5 96.6 62.7 55.4 59.5 53.4 54.4 59.3 57 61.4 62.3 67.8 66.6 81.5 64 53.9 57 47.8 49.8 58.9 56.5 63 62.8 68.4 71 89.9 64.7 55.7 57.8 52.6 53.8 57.7 57.7 64.9 67 70.1 68.9 88.3 62.4 58 58.9 53.5 52 60.9 58.5 64.7 70.7

73.5 73.5 92.7 63.5 55.6 56.5 52.6 54.3 56.2 58.2 66.2 69.3 72.4 74.5 102 63 55.9 57.4 52 52.5 60.9 65.4 72 72.3 76.1 83.7 114.8 72.5 59 62.4 55.5 59.9 68.9 62.9 74.2 75.7 84.2 88.1 122.4 80.6 64.5 64.2 57.9 62 72.9 73 78.9 80.3 89 95 130.7 82.9 68.5 72.2 61.3 63.7 75.1 76.9 83.6 80.9 96.5 102.9 141.2 91.4 72.9 79.2 70.8 77.9 86.1 83.7 91.7 93.6 104.2 113.5 154.9 106.6 79.6 88.1 77.4 84.2 93.6 98 108.2 96 105.3 112.5 147.5 104.1 76.8 81.7 71.1 74.9 95.4 89 107.5 93.7 102.2 111 151.2 102.2 85.1 89.2 76.5 83.3 94.5 95.1 107 95.4 103.5 111.8 156.2 106.4 81.5 88.2 77 79.8 94.6 92.8 111.7 93.4 107.6 119.7 155.3 118.3 87.4 90.7 78.5 83.7 98.8 95.8 106 97.3 110 116.2 143.1 114 81.5 85.1 78.9 84.2 97 92.9 112.5 100.4 114.9 121 152.9 126.3 86.3 95.1 82.5 86.7 108 102.9 113.2 103.4 117.2 118.1 148.7 119.8 88.7 92.1 87.8 94.8 112.2 109.1 127.4 110.3 120.4 127.8 152.7 126.1 91 105.9 92.1 101.1 113.9 113.6 130.1 116.3 127.7 140.9 164.7 129.4 103.7 113 103.2 111.3 131.1 127.1 138.3 125.4 132.5 145.9 169.5 132.4 111.7 117.6 106.9 113.1 128.8 122.2 142.2 122.5 130.5 140.4 154.3 .

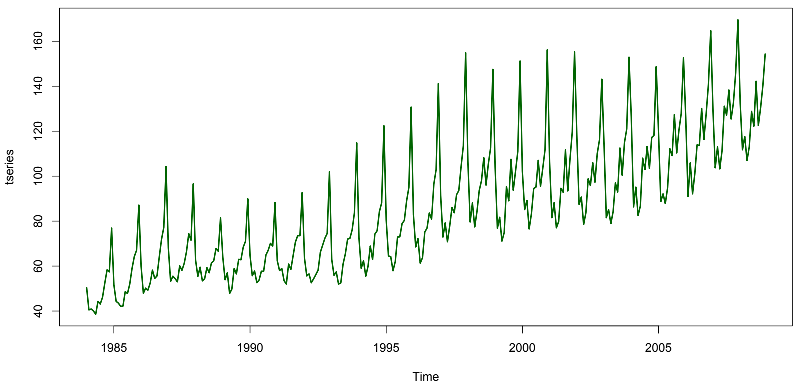
Anno	**Gen**	**Feb**	**Mar**	**Apr**	**Mag**	**Giu**	**Lug**	**Ago**	**Set**	**Ott**	**Nov**	**Dic**
1984	50.4	40.5	40.9	40	38.6	44.3	43.1	46.1	52.4	58.3	57.4	76.9
1985	51.7	44.3	43.5	42.1	42.2	48.6	47.8	52	58.9	64.2	67	87.1
1986	59.9	47.9	50.2	49.3	52.4	58.2	54.5	55.6	64	71.9	77.2	104.3
1987	68.1	53.2	55.5	54.4	53	60.1	58.1	61.2	66.4	74.4	71.5	96.6
1988	62.7	55.4	59.5	53.4	54.4	59.3	57	61.4	62.3	67.8	66.6	81.5
1989	64	53.9	57	47.8	49.8	58.9	56.5	63	62.8	68.4	71	89.9
1990	64.7	55.7	57.8	52.6	53.8	57.7	57.7	64.9	67	70.1	68.9	88.3
1991	62.4	58	58.9	53.5	52	60.9	58.5	64.7	70.7	73.5	73.5	92.7
1992	63.5	55.6	56.5	52.6	54.3	56.2	58.2	66.2	69.3	72.4	74.5	102
1993	63	55.9	57.4	52	52.5	60.9	65.4	72	72.3	76.1	83.7	114.8
1994	72.5	59	62.4	55.5	59.9	68.9	62.9	74.2	75.7	84.2	88.1	122.4
1995	80.6	64.5	64.2	57.9	62	72.9	73	78.9	80.3	89	95	130.7
1996	82.9	68.5	72.2	61.3	63.7	75.1	76.9	83.6	80.9	96.5	102.9	141.2
1997	91.4	72.9	79.2	70.8	77.9	86.1	83.7	91.7	93.6	104.2	113.5	154.9
1998	106.6	79.6	88.1	77.4	84.2	93.6	98	108.2	96	105.3	112.5	147.5
1999	104.1	76.8	81.7	71.1	74.9	95.4	89	107.5	93.7	102.2	111	151.2
2000	102.2	85.1	89.2	76.5	83.3	94.5	95.1	107	95.4	103.5	111.8	156.2
2001	106.4	81.5	88.2	77	79.8	94.6	92.8	111.7	93.4	107.6	119.7	155.3
2002	118.3	87.4	90.7	78.5	83.7	98.8	95.8	106	97.3	110	116.2	143.1
2003	114	81.5	85.1	78.9	84.2	97	92.9	112.5	100.4	114.9	121	152.9
2004	126.3	86.3	95.1	82.5	86.7	108	102.9	113.2	103.4	117.2	118.1	148.7
2005	119.8	88.7	92.1	87.8	94.8	112.2	109.1	127.4	110.3	120.4	127.8	152.7
2006	126.1	91	105.9	92.1	101.1	113.9	113.6	130.1	116.3	127.7	140.9	164.7
2007	129.4	103.7	113	103.2	111.3	131.1	127.1	138.3	125.4	132.5	145.9	169.5
2008	132.4	111.7	117.6	106.9	113.1	128.8	122.2	142.2	122.5	130.5	140.4	154.3

Nuovamente, da una parte abbiamo una serie già pronta per esser agilmente caricata su **R**, e d'altra parte abbiamo pure la sua forma più completa e chiara della quale conviene restar sempre ben consapevoli - già a cominciare dalla scelta dei due parametri di partenza `frequency` e `start` del codice.

Notiamo in particolare che tale serie esiste solo fino all'anno 2008 incluso, il quale è considerevolmente lontano dall'anno 2016 corrente; inoltre è ben possibile constatare che purtroppo, in modo analogo, ciò sussiste pure per tutte le altre serie disponibili sulle spese in arredamento: prima o poi dovremo necessariamente far pesare questo spiacevole difetto.

Comunque sia, evidenziamo lo stesso la serie, copiamola mediante ad esempio l'usuale combinazione di tasti `cmd+c`, e procediamo su **R** coi primi comandi ed il primo grafico a riguardo:

```
> X <- scan(pipe("pbpaste"))
Read 300 items
> tseries = ts(X,frequency=12,start=c(1984,1))
> ts.plot(tseries,lwd=2,col="darkgreen")
```



La serie così rappresentata, sebbene sembri leggermente disturbata da rumore di una certa importanza non del tutto trascurabile, parrebbe mostrare un chiaro trend ascendente generale ed in misura ancora maggiore una marcata stagionalità praticamente stazionaria e addirittura con periodo in accordo perfetto col parametro `frequency` preliminarmente scelto - periodo di dodici mesi o equivalentemente di un anno -, e semmai dei siffatti aspetti strutturali non dovrebbero risultare in verità affatto sorprendenti per la serie in questione, così come semmai una morbida ciclicità di fondo.

Andiamo naturalmente avanti con vere analisi matematiche tecniche a riguardo.

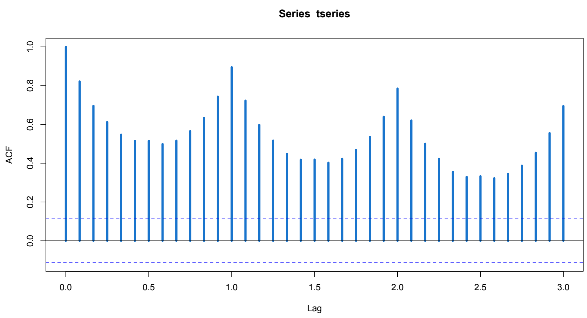
3.1 Funzione di auto-correlazione, decomposizioni della serie e analisi dei residui

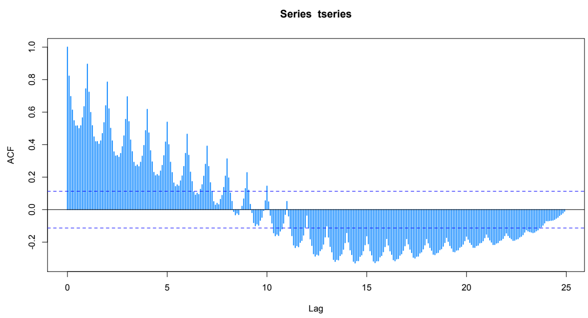
Teniamo tutta la serie e riprendiamo quindi i comandi digitati sopra.

3.1.1 Funzione di auto-correlazione

Cominciamo sempre dalla ACF su due lag estremali:

```
> acf(tseries,36,lwd=4,col="dodgerblue3")
> acf(tseries,300,lwd=2,col="dodgerblue")
```



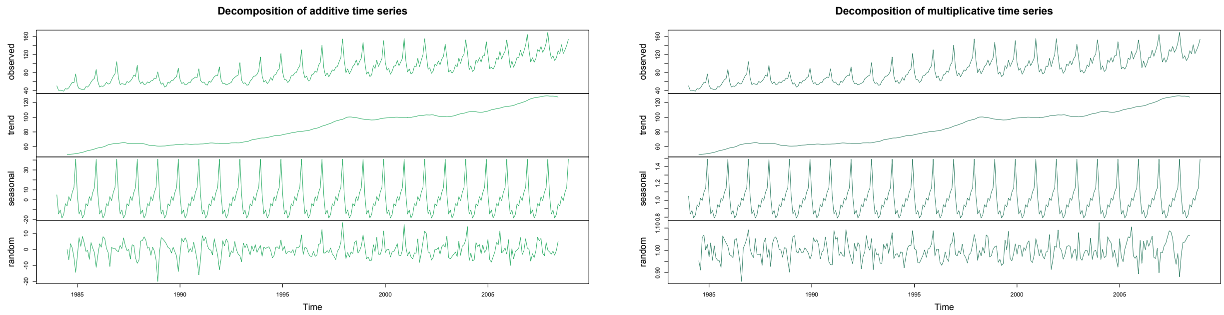


Del tutto perfetto: teniamo il `frequency=12`.

3.1.2 Decomposizioni della serie

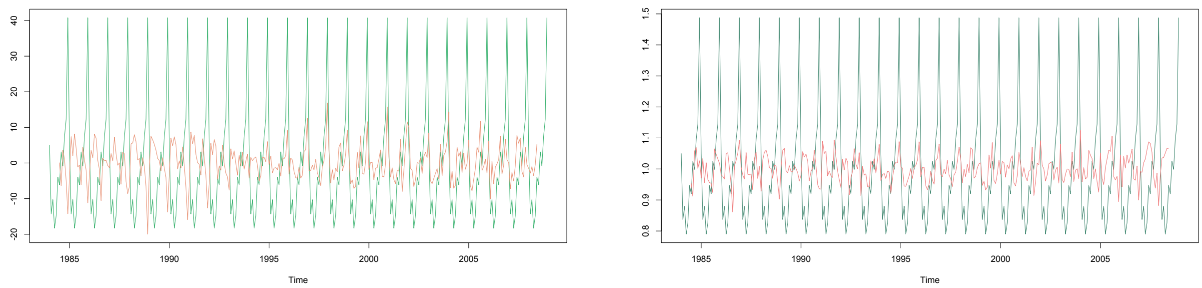
Decomposizione additiva e decomposizione moltiplicativa rispettivamente:

```
> dec.add = decompose(tseries)
> dec.mult = decompose(tseries,type="multiplicative")
> plot(dec.add,col="mediumseagreen")
> plot(dec.mult,col="aquamarine4")
```



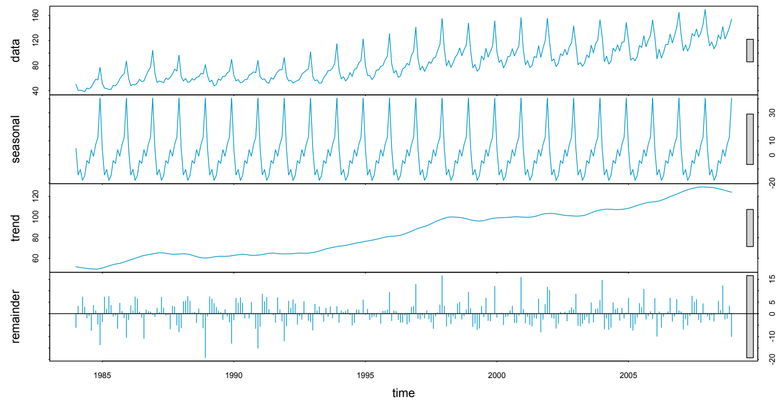
Ecco stavolta che in effetti una stagionalità vera e propria c'è, assieme al trend, e come: da una parte, infatti,

```
> ts.plot(dec.add$seasonal,dec.add$random,col=c("mediumseagreen","darksalmon"))
> ts.plot(dec.mult$seasonal,dec.mult$random,col=c("aquamarine4","lightcoral"))
```



D'altra parte, ancor meglio,

```
> plot(stl(tseries,300),col="deepskyblue3")
```



In definitiva, è vero che la serie è costituita principalmente di trend tanto quanto è vero però che va evolvendosi con una reale stagionalità la quale non è da meno, e la quale tra l'altro tende via via a restare pressoché stazionaria: cerchiamo dunque di capire quale delle due decomposizioni la spieghi meglio, senza ricorrere invece all'ausilio di opportuni `stl` - che invece, in linea di principio, avrebbero potuto rivelarsi particolarmente utili per una serie come questa.

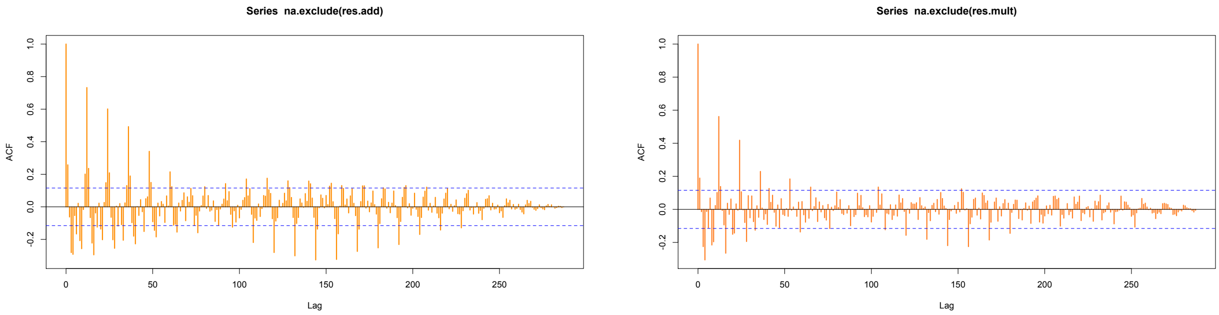
3.1.3 Analisi dei residui

Omogeneizziamo le due serie dei residui e confrontiamole poi a dovere:

```
> res.add = dec.add$random
> res.mult = dec.mult$random
> trend.mult = dec.mult$trend
> res.mult = mean(na.exclude(trend.mult)) * (res.mult - 1)
```

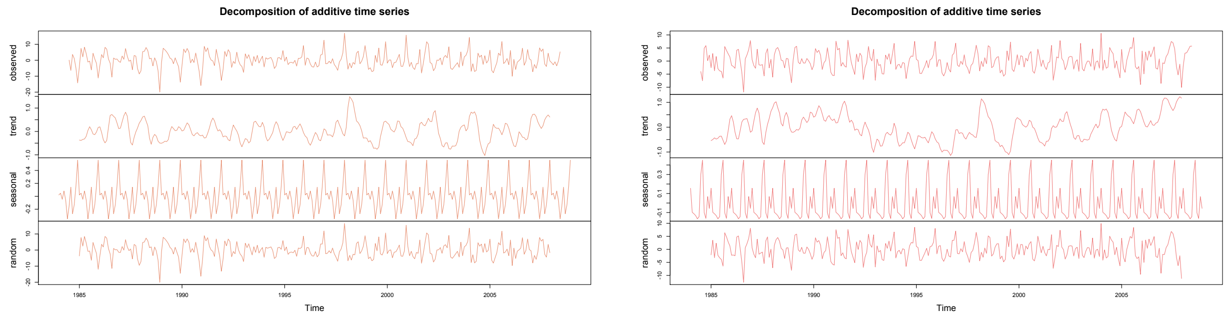
Rispettive funzioni di auto-correlazione:

```
> acf(na.exclude(res.add),300,lwd=2,col="darkorange")
> acf(na.exclude(res.mult),300,lwd=2,col="chocolate1")
```



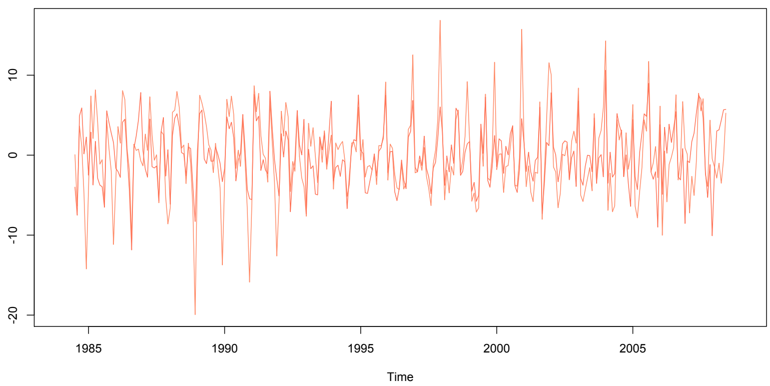
Rispettive decomposizioni additive:

```
> plot(decompose(res.add),col="darksalmon")
> plot(decompose(res.mult),col="lightcoral")
```



Le due serie semplicemente disegnate assieme:

```
> ts.plot(res.add,res.mult,col=c("salmon1","coral1"))
```



Rispettive varianze spiegate calcolate tenendo conto dei rispettivi dati `NA` presenti:

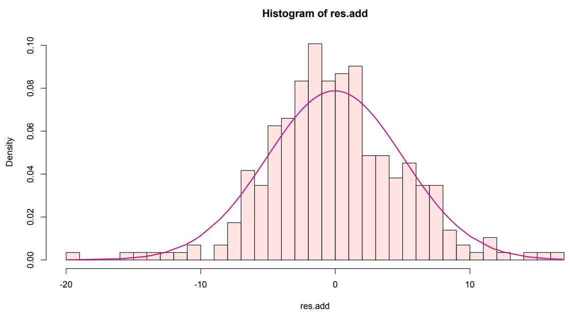
```
> 1 - var(na.exclude(res.add))/var(tseries[7:294])
[1] 0.9660886
> 1 - var(na.exclude(res.mult))/var(tseries[7:294])
[1] 0.9811082
```

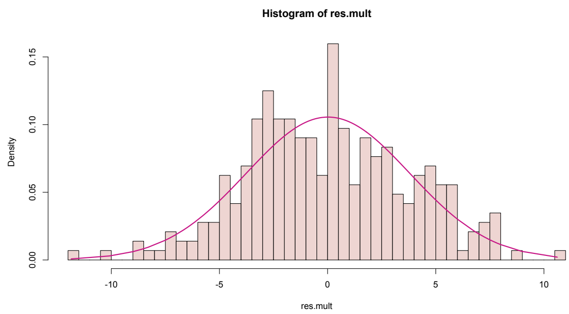
Solita nota a riguardo della seconda serie dei residui:

```
> 1 - var(na.exclude(log(dec.mult$random)))/var(tseries[7:294])
[1] 0.9999974
```

Rispettive medie ed istogrammi:

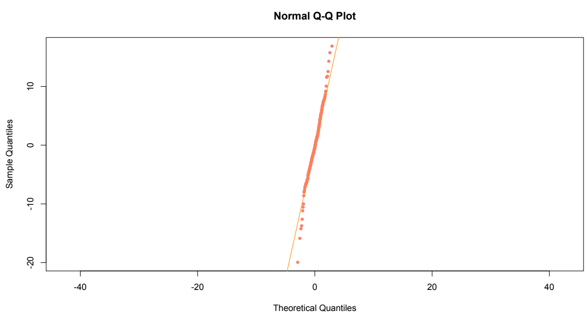
```
> res.add.s = sort(na.exclude(res.add))
> Z_1 = dnorm(res.add.s,mean(res.add.s),sd(res.add.s))
> mean(res.add.s)
[1] -0.0239728
> res.mult.s = sort(na.exclude(res.mult))
> Z_2 = dnorm(res.mult.s,mean(res.mult.s),sd(res.mult.s))
> mean(res.mult.s)
[1] 0.01805771
> hist(res.add,30,freq=FALSE,col="mistyrose1")
> lines(res.add.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.mult,35,freq=FALSE,col="mistyrose2")
> lines(res.mult.s,Z_2,lwd=2,col="mediumvioletred")
```

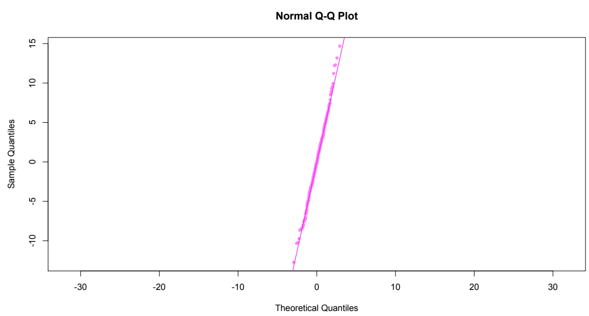




Rispettivi quantili, finalmente: per la prima,

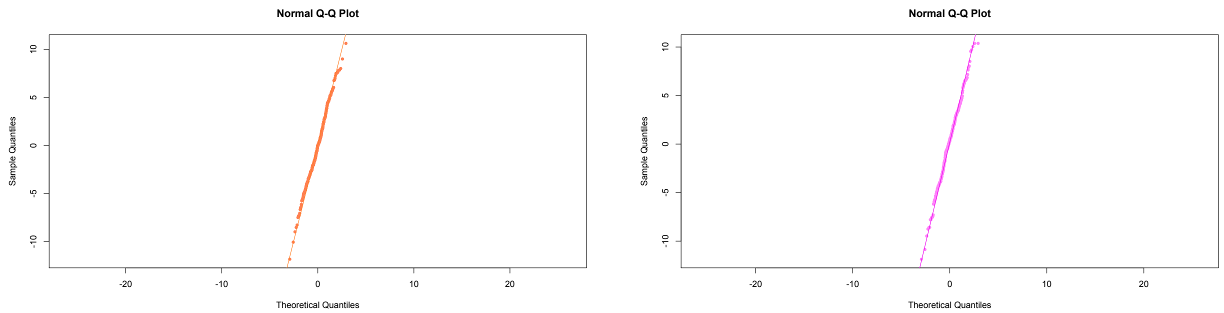
```
> qqnorm(res.add.s,pch=20,col="salmon",asp=1)
> qqline(res.add.s,col="darkorange")
> Z_1 = rnorm(length(res.add.s),mean(res.add.s),sd(res.add.s))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```





Per la seconda,

```
> qqnorm(res.mult.s,pch=20,col="coral",asp=1)
> qqline(res.mult.s,col="chocolate1")
> Z_2 = rnorm(length(res.mult.s),mean(res.mult.s),sd(res.mult.s))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```

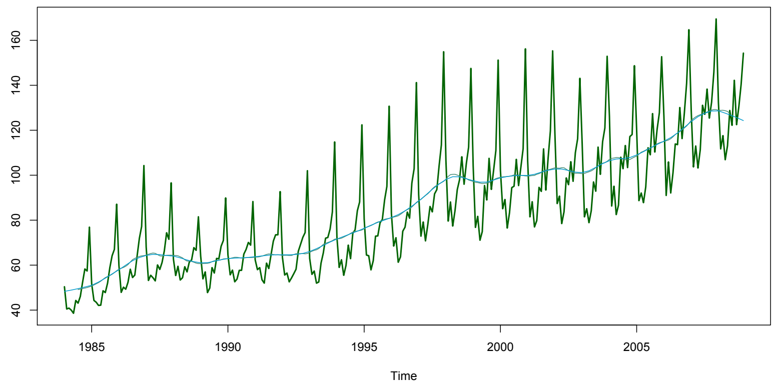


Visto tutto, affermiamo che entrambe le decomposizioni sono sicuramente piuttosto soddisfacenti, apprezzando in particolare i due `qqnorm` dell'analisi dei relativi residui, ma concludiamo che la decomposizione moltiplicativa è senz'altro preferibile a quella additiva, in questo caso.

3.2 Analisi e previsioni del trend e della serie

Continuiamo a considerare opportuna una previsione dell'andamento della serie in un prossimo futuro - anno 2017 - la quale sia il risultato del confronto di analisi e previsione del solo trend con analisi e previsione di un filtraggio della serie il quale sia una specie di via di mezzo proprio tra trend e serie stessa. Cominciamo con calma tracciando insieme la serie, il suo trend ed anche uno smoothing del medesimo per poi analizzarli e quindi prevederli separatamente uno ad uno, convenendo naturalmente di esser ben più accurati nel farlo per l'intera serie:

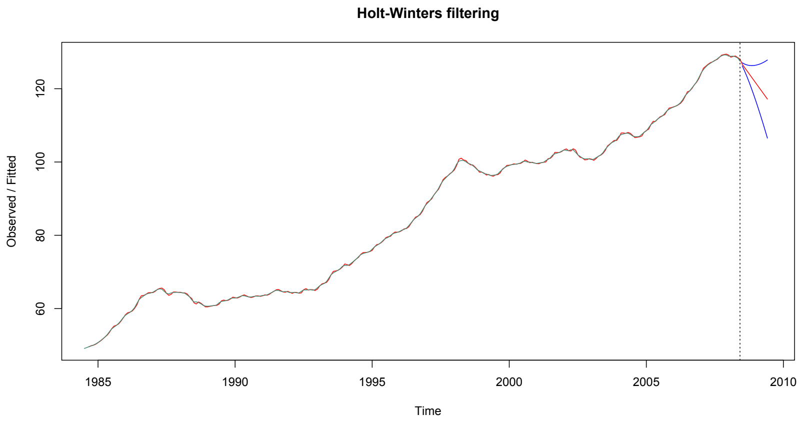
```
> ts.plot(tseries,trend.mult,stl(tseries,6)$time.series[,2],lwd=c(2,1,1),
+ col=c("darkgreen","aquamarine4","deepskyblue3"))
```



3.2.1 Analisi e previsioni del trend

Partiamo da un SET automatico per il trend:

```
> trend = trend.mult
> trend.SET = HoltWinters(na.omit(trend),gamma=FALSE)
> plot(trend.SET,predict(trend.SET,12,prediction.interval=TRUE),col="aquamarine4")
```

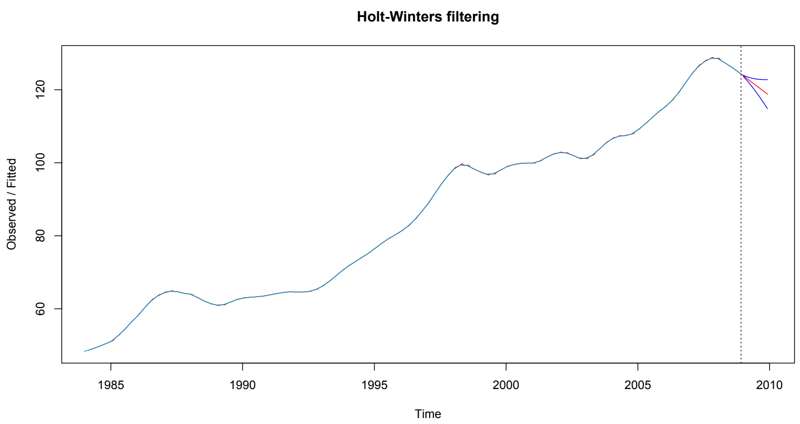


Una nota:

```
> trend.SET

Smoothing parameters:
 alpha: 1
 beta : 1
 gamma: FALSE
```

Guardiamo quindi il SET automatico per lo smoothing:

```
> trend = stl(tseries,6)$time.series[,2]
> trend.SET = HoltWinters(trend,gamma=FALSE)
> plot(trend.SET,predict(trend.SET,12,prediction.interval=TRUE),col="deepskyblue3")
```

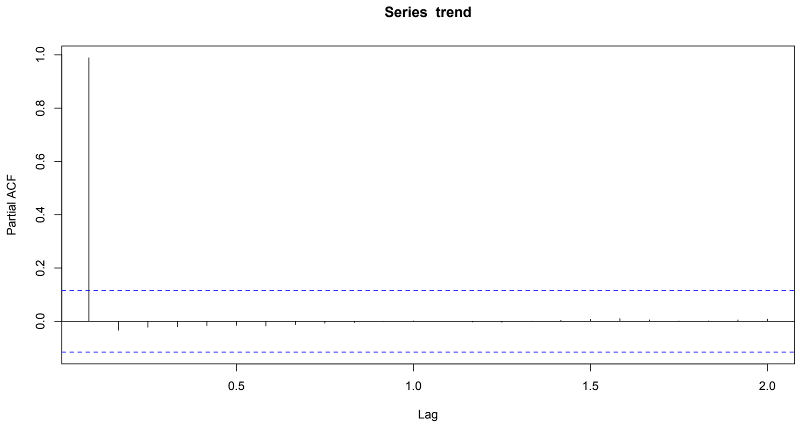


Di nuovo:

```
> trend.SET

Smoothing parameters:
 alpha: 1
 beta : 1
 gamma: FALSE
```

Procediamo adesso con un'auto-regressione lineare effettuata manualmente per il trend, basandosi sulla funzione di auto-correlazione parziale per il trend stesso:

```
> trend = na.omit(trend.mult)
> pacf(trend)
```



Costruiamo un modello lineare per il trend:

```
> n = length(trend)
> k = 12
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,12]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7]+A[,8]+A[,9]+A[,10]
+ +A[,11])
> summary(trend.lm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04346    0.04465   0.974 0.331181
A[, 1]       0.10368    0.06323   1.640 0.102252
A[, 2]      -0.34262    0.15904  -2.154 0.032117 *
A[, 3]       0.72427    0.21292   3.402 0.000773 ***
A[, 4]      -1.04861    0.23856  -4.396 1.60e-05 ***
A[, 5]       1.14222    0.25235   4.526 9.07e-06 ***
A[, 6]      -1.28616    0.25574  -5.029 9.09e-07 ***
A[, 7]       1.48236    0.25108   5.904 1.08e-08 ***
A[, 8]      -1.75163    0.23690  -7.394 1.87e-12 ***
A[, 9]       2.00370    0.21117   9.489  < 2e-16 ***
A[, 10]     -2.36788    0.15740 -15.044  < 2e-16 ***
A[, 11]      2.34057    0.06230  37.572  < 2e-16 ***

Residual standard error: 0.1793 on 265 degrees of freedom
Multiple R-squared:  0.9999,Adjusted R-squared:  0.9999
F-statistic: 3.678e+05 on 11 and 265 DF,  p-value: < 2.2e-16
```

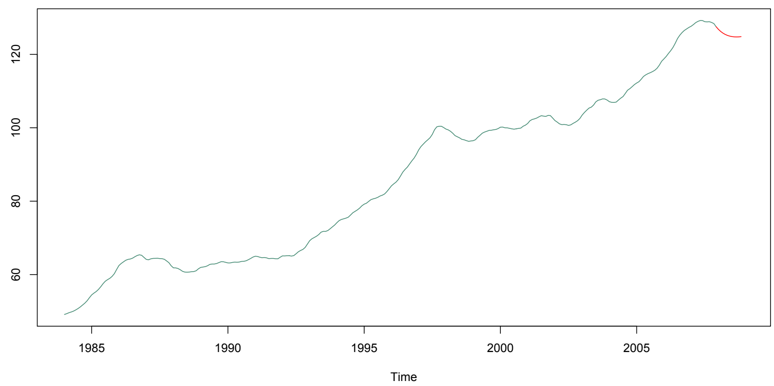
Ottimizziamo tale modello:

```
> trend.lm = lm(A[,12]~A[,10]+A[,11]+0)
> summary(trend.lm)

Coefficients:
```

```
        Estimate Std. Error t value Pr(>|t|)
A[, 10] -0.86745    0.03222  -26.93   <2e-16 ***
A[, 11]  1.86777    0.03212   58.16   <2e-16 ***

Residual standard error: 0.2103 on 275 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 2.511e+07 on 2 and 275 DF,  p-value: < 2.2e-16
```

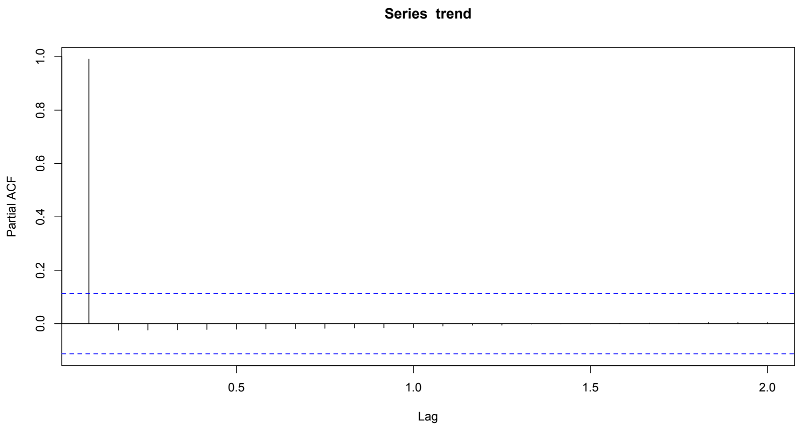
Conseguente previsione per il trend:

```
> pred = 1:(n+k-1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+12)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> trend.P = ts(trend.P,frequency=12,start=c(1984,1))
> pred.P = ts(pred.P,frequency=12,start=c(1984,1))
> ts.plot(trend.P,pred.P,col=c("aquamarine4","red"))
```



Volendo ora far lo stesso per lo smoothing, guardiamo la sua funzione di auto-correlazione parziale:

```
> trend = stl(tseries,6)$time.series[,2]
> pacf(trend)
```



Costruiamo un modello lineare per lo smoothing:

```
> n = length(trend)
> k = 12
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = trend[j:(n-k+j)]
+ }
> trend.lm = lm(A[,12]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7]+A[,8]+A[,9]+A[,10]
+ +A[,11])
> summary(trend.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01869    0.01276   1.464   0.1443
A[, 1]      -0.04853    0.06109  -0.794   0.4276
A[, 2]       0.13635    0.13266   1.028   0.3049
A[, 3]      -0.08728    0.14422  -0.605   0.5455
A[, 4]      -0.39053    0.15331  -2.547   0.0114 *
A[, 5]       0.75823    0.18377   4.126 4.89e-05 ***
A[, 6]      -0.36886    0.19026  -1.939   0.0536 .
A[, 7]       0.91621    0.18350   4.993 1.05e-06 ***
A[, 8]      -1.84912    0.15299 -12.086  < 2e-16 ***
A[, 9]       0.93423    0.14419   6.479 4.20e-10 ***
A[, 10]     -0.96387    0.13209  -7.297 3.12e-12 ***
A[, 11]      1.96306    0.06003  32.699  < 2e-16 ***

Residual standard error: 0.05326 on 277 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 4.669e+06 on 11 and 277 DF,  p-value: < 2.2e-16
```

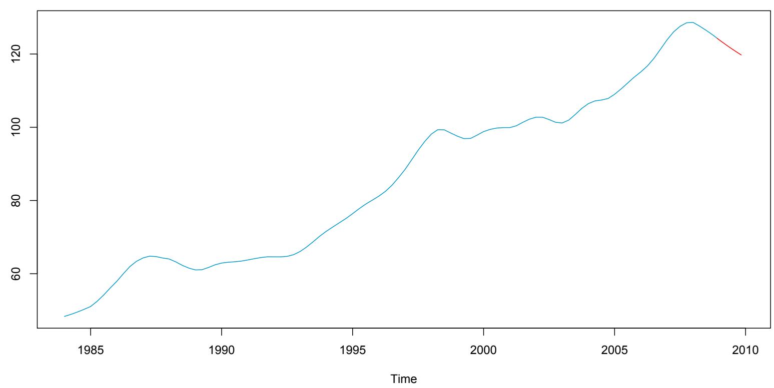
Ottimizziamo tale modello:

```
> trend.lm = lm(A[,12]~A[,10]+A[,11]+0)
> summary(trend.lm)

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
A[, 10] -0.98240    0.01466  -66.99   <2e-16 ***
A[, 11]  1.98240    0.01462  135.57   <2e-16 ***
```

```
Residual standard error: 0.08038 on 287 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 1.802e+08 on 2 and 287 DF,  p-value: < 2.2e-16
```

Conseguente previsione per lo smoothing:

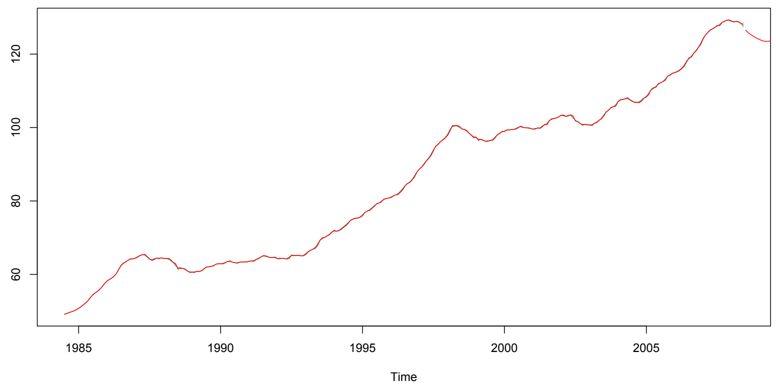
```
> pred = 1:(n+k-1)
> pred[1:n] = trend
> for(i in 1:(k-1)){
+ pred[n+i] = coef(trend.lm) %*% c(pred[n+i-2],pred[n+i-1])
+ }
> trend.P = pred.P = pred
> for(i in (n+1):(n+12)){
+ trend.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> trend.P = ts(trend.P,frequency=12,start=c(1984,1))
> pred.P = ts(pred.P,frequency=12,start=c(1984,1))
> ts.plot(trend.P,pred.P,col=c("deepskyblue3","red"))
```



Affianchiamo al tutto pure un metodo automatico dei minimi quadrati nell'auto-regressione lineare per il trend, e solo per questo:

```
> trend = na.omit(trend.mult)
> trend.ar = ar(trend,method="ols",aic=TRUE)

> trend.ar

Order selected 24  sigma^2 estimated as  0.02625

> 1 - trend.ar$var/var(trend)
          [,1]
[1,] 0.9999477

> o = trend.ar$order
> a = trend.ar$ar
> trend.ar.P = trend
> for(i in (o+1):length(trend)){
```

```
+ trend.ar.P[i] = sum(rev(a)*trend[(i-o):(i-1)]) + mean(trend)*(1-sum(a))
+ }
> ts.plot(trend,trend.ar.P,col=c("aquamarine4","red"))
> lines(predict(trend.ar,n.ahead=12,se.fit=FALSE),col="red")
```



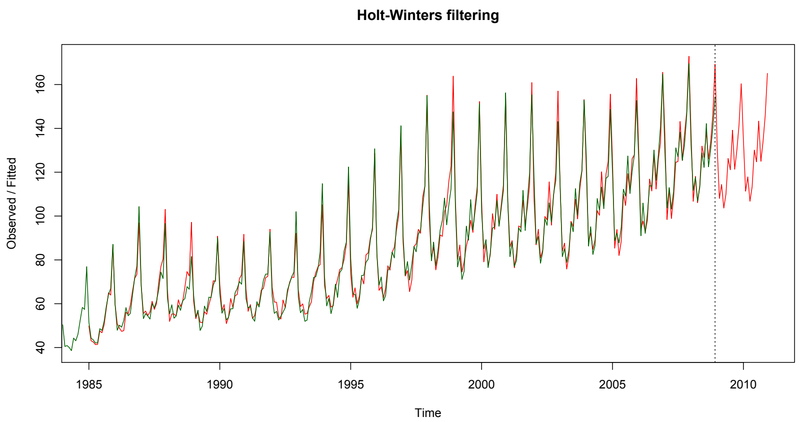
Di nuovo siamo più contenti dell'auto-regressione lineare svolta per il trend e per lo smoothing scelto, specie per il trend, in quanto sembra confermare una tendenza del trend stesso ad un'ascesa generale a fronte di eventuali ma leggere discese, magari più istantanee che non, e questo in accordo con una certa ciclicità di fondo nella quale tendiamo davvero a credere.

Di conseguenza ci sentiamo fiduciosi in un'auto-regressione lineare anche per l'analisi e la previsione dell'intera serie, sperando logicamente in delle conclusioni analoghe, senza però aspettarsi affatto una prestazione deludente da parte del metodo di Holt-Winters il quale al contrario risulta di solito particolarmente efficace proprio sulle serie di stagionalità pronunciata.

3.2.2 Analisi e previsioni della serie

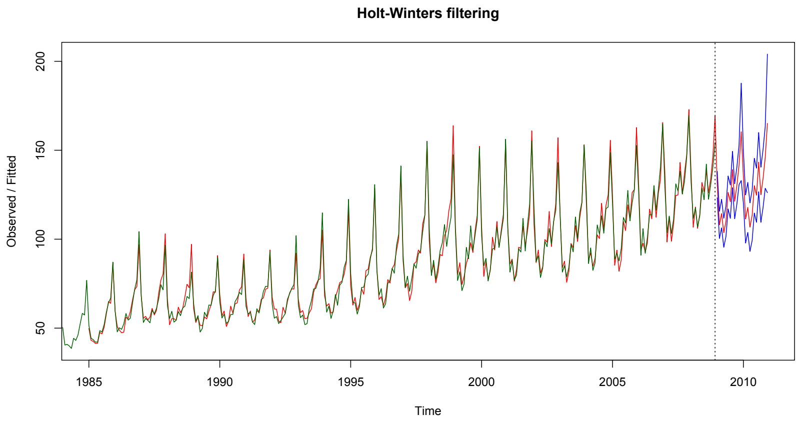
Per cominciare, proponiamo giustappunto un Holt-Winters (HW) automatico ma che sia forzato a dipendere in modo essenziale dalla decomposizione moltiplicativa della serie:

```
> tseries.HW = HoltWinters(tseries,seasonal="multiplicative")
> plot(tseries.HW,predict(tseries.HW,24),col="darkgreen")
```



Ci convinciamo abbastanza in fretta che un tale risultato sia già più che buono per la nostra serie, e allora lo teniamo e proviamo a compiere un'innocua aggiunta al precedente grafico:

```
> plot(tseries.HW,predict(tseries.HW,24,prediction.interval=TRUE),col="darkgreen")
```

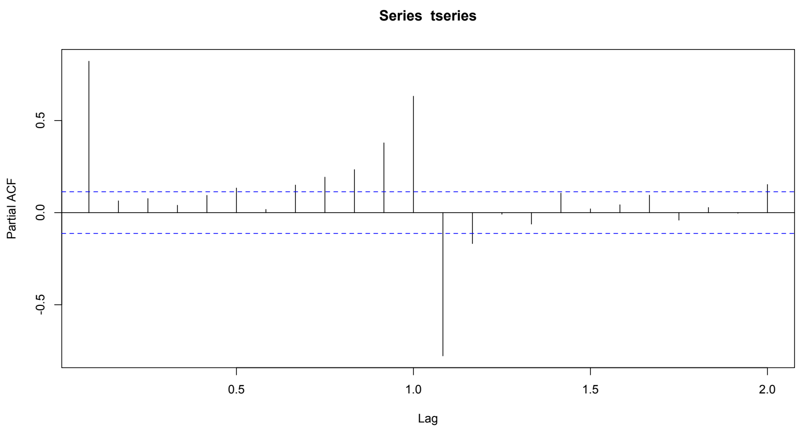


Riportiamo inoltre i tre parametri fondamentali di questo HW:

```
> tseries.HW

Smoothing parameters:
 alpha: 0.2704941
 beta : 0.002780534
 gamma: 0.8825925
```

Andiamo ora di auto-regressione lineare manuale partendo dalla funzione di auto-correlazione parziale:

```
> pacf(tseries)
```



Costruiamo un modello lineare:

```
> n = length(tseries)
> k = 12
> A = matrix(nrow=n-k+1,ncol=k)
> for(j in 1:k){
+ A[,j] = tseries[j:(n-k+j)]
+ }
> tseries.lm = lm(A[,12]~A[,1]+A[,2]+A[,3]+A[,4]+A[,5]+A[,6]+A[,7]+A[,8]+A[,9]+A[,10]
+ +A[,11])
> summary(tseries.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

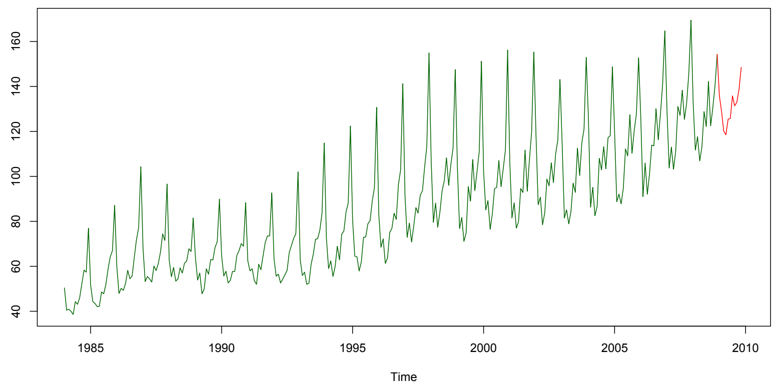
```
(Intercept)  3.268921   3.021300   1.082   0.2802
A[, 1]        0.468045   0.053992   8.669 3.73e-16 ***
A[, 2]       -0.008409   0.061881  -0.136   0.8920
A[, 3]        0.023335   0.061835   0.377   0.7062
A[, 4]       -0.015600   0.061847  -0.252   0.8010
A[, 5]       -0.049074   0.061808  -0.794   0.4279
A[, 6]        0.115258   0.061422   1.876   0.0616 .
A[, 7]       -0.042512   0.061686  -0.689   0.4913
A[, 8]       -0.033255   0.061397  -0.542   0.5885
A[, 9]        0.033908   0.061225   0.554   0.5801
A[, 10]      -0.004914   0.061232  -0.080   0.9361
A[, 11]       0.494218   0.053556   9.228  < 2e-16 ***

Residual standard error: 13.06 on 277 degrees of freedom
Multiple R-squared:  0.791,Adjusted R-squared:  0.7827
F-statistic: 95.32 on 11 and 277 DF,  p-value: < 2.2e-16
```

Ottimizziamo tale modello:

```
> tseries.lm = lm(A[,12]~A[,1]+A[,11]+0)
> summary(tseries.lm)

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
A[, 1]   0.49124    0.03562   13.79   <2e-16 ***
A[, 11]  0.52422    0.03462   15.14   <2e-16 ***

Residual standard error: 12.98 on 287 degrees of freedom
Multiple R-squared:   0.98,Adjusted R-squared:  0.9799
F-statistic:  7033 on 2 and 287 DF,  p-value: < 2.2e-16
```

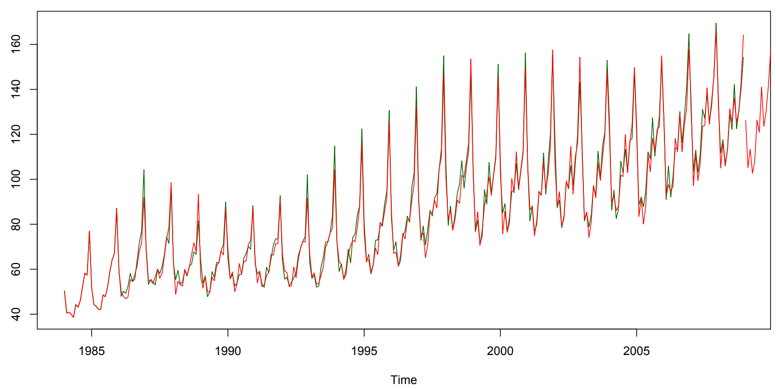
Conseguente previsione:

```
> pred = 1:(n+k-1)
> pred[1:n] = tseries
> for(i in 1:(k-1)){
+ pred[n+i] = coef(tseries.lm) %*% c(pred[n+i-11],pred[n+i-1])
+ }
> tseries.P = pred.P = pred
> for(i in (n+1):(n+12)){
+ tseries.P[i] = NA
+ }
> for(i in 1:(n-1)){
+ pred.P[i] = NA
+ }
> tseries.P = ts(tseries.P,frequency=12,start=c(1984,1))
> pred.P = ts(pred.P,frequency=12,start=c(1984,1))
> ts.plot(tseries.P,pred.P,col=c("darkgreen","red"))
```



Occhiata al metodo automatico dei minimi quadrati nell'auto-regressione lineare:

```
> tseries.ar = ar(tseries,method="ols",aic=TRUE)

> tseries.ar

Order selected 24  sigma^2 estimated as  14.99

> 1 - tseries.ar$var/var(tseries)
          [,1]
[1,] 0.9816167

> o = tseries.ar$order
> a = tseries.ar$ar
> tseries.ar.P = tseries
> for(i in (o+1):length(tseries)){
+ tseries.ar.P[i] = sum(rev(a)*tseries[(i-o):(i-1)]) + mean(tseries)*(1-sum(a))
+ }
> ts.plot(tseries,tseries.ar.P,col=c("darkgreen","red"))
> lines(predict(tseries.ar,n.ahead=12,se.fit=FALSE),col="red")
```



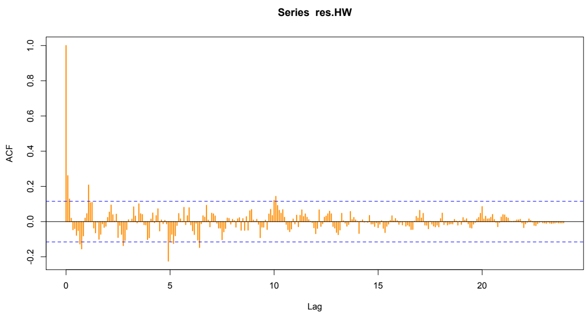
E così è andato tutto più che bene, e anzi saremmo tentati di giudicare l'analisi di HW come la preferibile, ma passiamo ovviamente ai residui delle analisi.

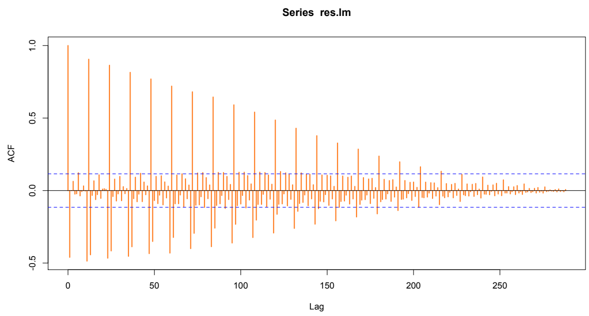
3.3 Analisi dei residui ed incertezza delle previsioni

3.3.1 Analisi dei residui

Estraiamo le due serie dei residui ed osserviamo le rispettive funzioni di auto-correlazione:

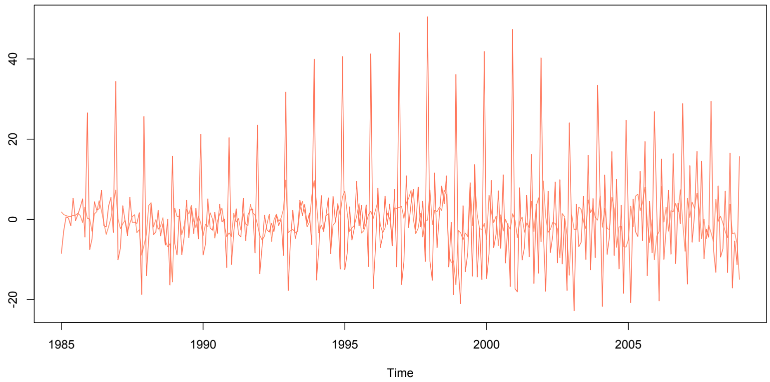
```
> res.HW = residuals(tseries.HW)
> length(res.HW)
[1] 288
> res.lm = residuals(tseries.lm)
> length(res.lm)
[1] 289
> acf(res.HW,290,lwd=2,col="darkorange")
> acf(res.lm,290,lwd=2,col="chocolate1")
```





La seconda è davvero inammissibile!! In effetti, disegnando assieme le due serie,

```
> res.lm = ts(res.lm,frequency=12,start=c(1984,12))
> ts.plot(res.HW,res.lm[-1],col=c("salmon1","coral1"))
```



Oppure, le rispettive varianze spiegate dopo esserci chiariti le idee sulla precisa forma dei residui:

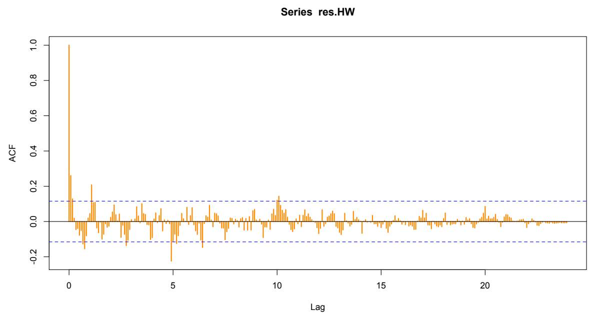
```
> 1 - var(res.HW)/var(tseries[13:300])
[1] 0.978155
> 1 - var(res.lm)/var(tseries[12:300])
[1] 0.7862155
```

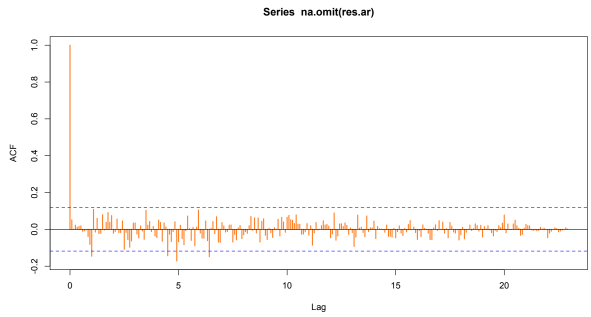
Sostituiamo la seconda serie a quella ottenuta dal metodo automatico dei minimi quadrati:

```
> res.ar = tseries.ar$resid
> length(na.omit(res.ar))
[1] 276
```

Relativa funzione di auto-correlazione affiancata a quella della prima serie:

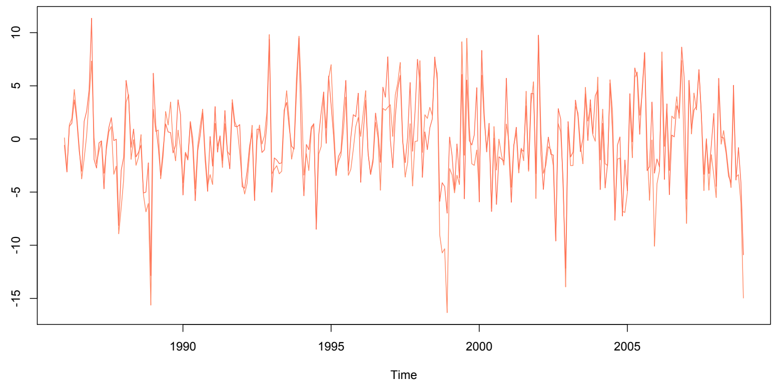
```
> acf(na.omit(res.ar),280,lwd=2,col="chocolate1")
```





Disegnamo la prima serie e quest'ultima assieme:

```
> ts.plot(res.HW[13:288],na.omit(res.ar),col=c("salmon1","coral1"))
```

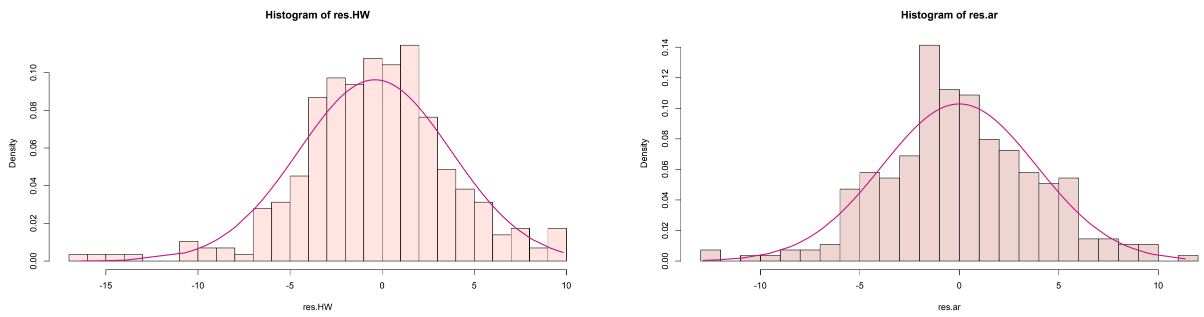


Calcoliamo le rispettive varianze spiegate dopo esserci chiariti le idee sulla precisa forma dei residui:

```
> 1 - var(res.HW)/var(tseries[13:300])
[1] 0.978155
> 1 - var(na.omit(res.ar))/var(tseries[25:300])
[1] 0.9803187
```

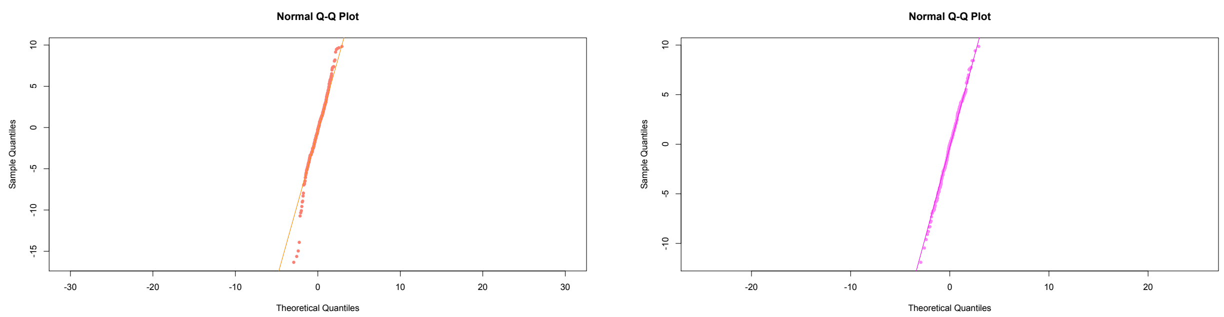
Addirittura un piccolo sorpasso! Comunque sia, rispettive medie ed istogrammi:

```
> res.HW.s = sort(res.HW)
> Z_1 = dnorm(res.HW.s,mean(res.HW),sd(res.HW))
> mean(res.HW)
[1] -0.3941146
> res.ar.s = sort(na.exclude(res.ar))
> Z_2 = dnorm(res.ar.s,mean(res.ar.s),sd(res.ar.s))
> mean(res.ar.s)
[1] -4.321473e-14
> hist(res.HW,20,freq=FALSE,col="mistyrose1")
> lines(res.HW.s,Z_1,lwd=2,col="mediumvioletred")
> hist(res.ar,20,freq=FALSE,col="mistyrose2")
> lines(res.ar.s,Z_2,lwd=2,col="mediumvioletred")
```



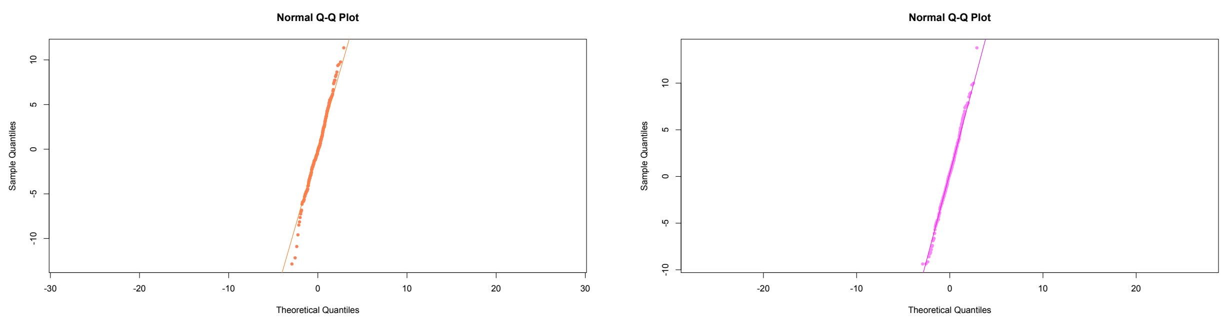
Infine i rispettivi quantili: per la prima,

```
> qqnorm(res.HW,pch=20,col="salmon",asp=1)
> qqline(res.HW,col="darkorange")
> Z_1 = rnorm(length(res.HW),mean(res.HW),sd(res.HW))
> qqnorm(Z_1,pch=20,col="orchid1",asp=1)
> qqline(Z_1,col="magenta2")
```



Per l'altra,

```
> qqnorm(res.ar.s,pch=20,col="coral",asp=1)
> qqline(res.ar.s,col="chocolate1")
> Z_2 = rnorm(length(res.ar.s),mean(res.ar.s),sd(res.ar.s))
> qqnorm(Z_2,pch=20,col="orchid1",asp=1)
> qqline(Z_2,col="magenta2")
```



Visto tutto, i due metodi ci sembrano simili ed entrambi davvero ben riusciti, pensando il secondo forse di poco più positivo, ma tuttavia veniamo ora al finale portandoci dietro entrambe le serie di residui.

3.3.2 Incertezza delle previsioni

Anzitutto, due quantili per entrambe: per la prima,

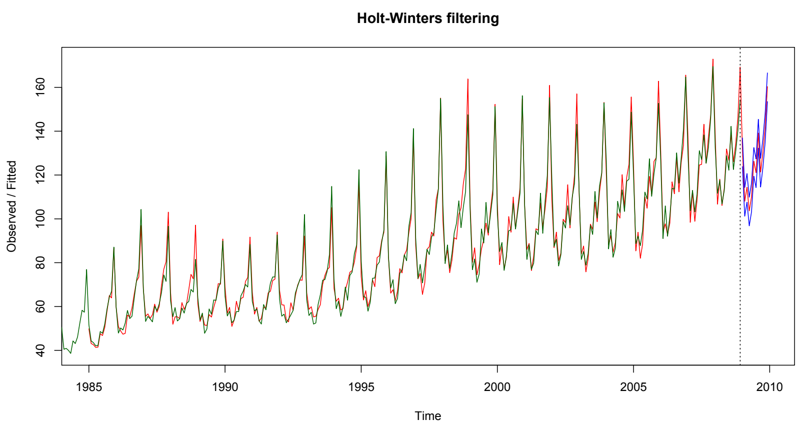
```
> quantile(res.HW,0.05)
       5%
-6.838083
> quantile(res.HW,0.95)
     95%
6.250626
> qnorm(0.05,mean(res.HW),sd(res.HW))
[1] -7.214165
> qnorm(0.95,mean(res.HW),sd(res.HW))
[1] 6.425936
```

Per la seconda,

```
> quantile(res.ar.s,0.05)
       5%
-5.822828
> quantile(res.ar.s,0.95)
     95%
6.096576
> qnorm(0.05,mean(res.ar.s),sd(res.ar.s))
[1] -6.380878
> qnorm(0.95,mean(res.ar.s),sd(res.ar.s))
[1] 6.380878
```

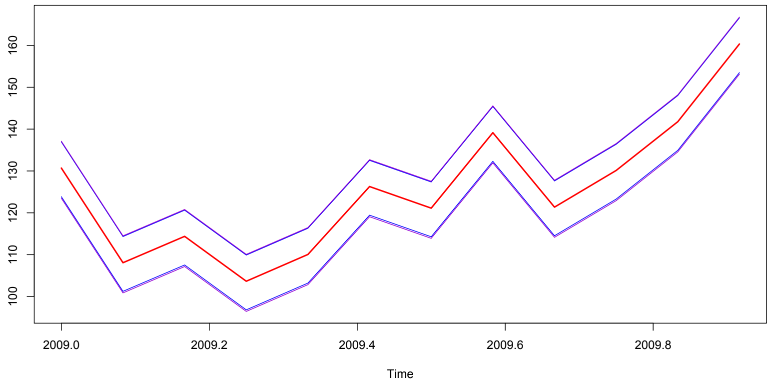
Quindi una previsione con banda di confidenza non parametrica per la prima:

```
> plot(tseries.HW,predict(tseries.HW,12),col="darkgreen")
> lines(predict(tseries.HW,12) + quantile(res.HW,0.05),col="blue")
> lines(predict(tseries.HW,12) + quantile(res.HW,0.95),col="blue")
```



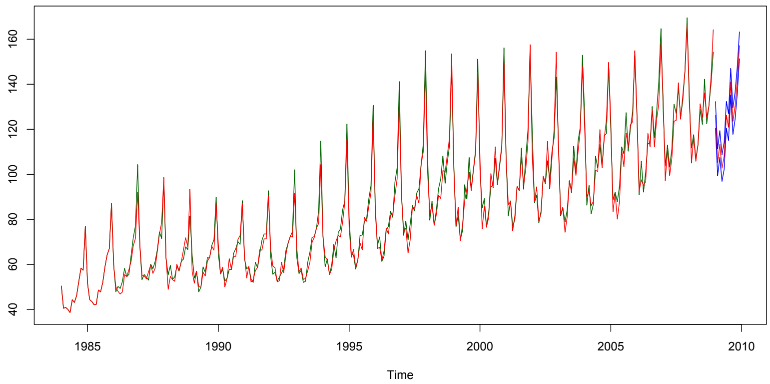
Poi un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica:

```
> ts.plot(predict(tseries.HW,12),
+ predict(tseries.HW,12) + quantile(res.HW,0.05),
+ predict(tseries.HW,12) + quantile(res.HW,0.95),
+ predict(tseries.HW,12) + qnorm(0.05,mean(res.HW),sd(res.HW)),
+ predict(tseries.HW,12) + qnorm(0.95,mean(res.HW),sd(res.HW)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



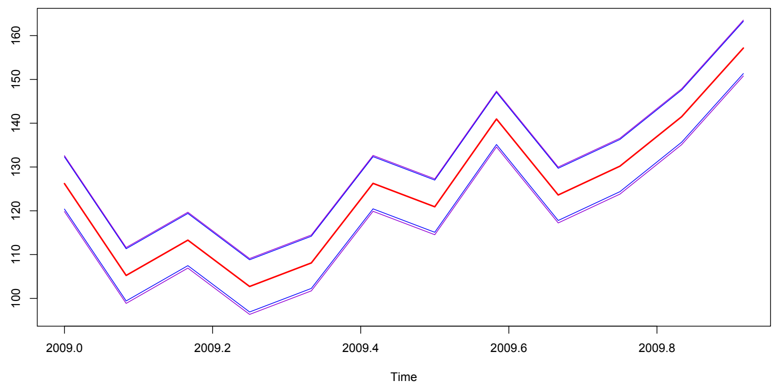
Adesso una previsione con banda di confidenza non parametrica per la seconda:

```
> ts.plot(tseries,tseries.ar.P,predict(tseries.ar,n.ahead=12,se.fit=FALSE),
+ col=c("darkgreen","red","red"))
> lines(predict(tseries.ar,n.ahead=12,se.fit=FALSE) + quantile(res.ar.s,0.05),col="blue")
> lines(predict(tseries.ar,n.ahead=12,se.fit=FALSE) + quantile(res.ar.s,0.95),col="blue")
```



Infine un relativo ingrandimento con l'aggiunta della banda di confidenza parametrica:

```
> ts.plot(predict(tseries.ar,n.ahead=12,se.fit=FALSE),
+ predict(tseries.ar,n.ahead=12,se.fit=FALSE) + quantile(res.ar.s,0.05),
+ predict(tseries.ar,n.ahead=12,se.fit=FALSE) + quantile(res.ar.s,0.95),
+ predict(tseries.ar,n.ahead=12,se.fit=FALSE) + qnorm(0.05,mean(res.ar.s),sd(res.ar.s)),
+ predict(tseries.ar,n.ahead=12,se.fit=FALSE) + qnorm(0.95,mean(res.ar.s),sd(res.ar.s)),
+ lwd=c(2,1,1,1,1),col=c("red","blue","blue","darkviolet","darkviolet"))
```



Di nuovo un'altra innegabile simiglianza, ma di nuovo sbilanciata forse a favore dell'auto-regressione.

Pensiamo pertanto che sia proprio il caso di provare un'autovalidazione dei due metodi sull'anno 2008, ossia sulla seguente serie:

132.4 111.7 117.6 106.9 113.1 128.8 122.2 142.2 122.5 130.5 140.4 154.3 .

Dunque, per procedere, `cmd+c` e solito codice:

```
> x <- scan(pipe("pbpaste"))
Read 12 items
> x = ts(x,frequency=12,start=c(2008,1))
```

Oppure, direttamente:

```
> x = window(tseries,2008)
```

Metodo di Holt-Winters:

```
> tseries.x = window(tseries,1984,c(2007,12))
> tseries.x.HW = HoltWinters(tseries.x,seasonal="multiplicative")
```

Metodo dei minimi quadrati:

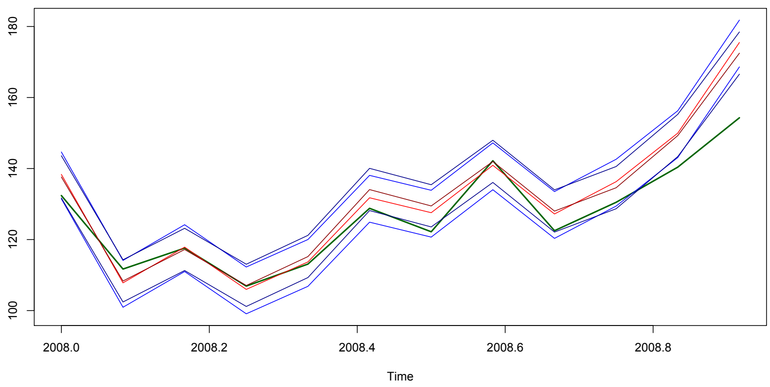
```
> tseries.x.ar = ar(tseries.x,method="ols",aic=TRUE)
> o = tseries.x.ar$order
> a = tseries.x.ar$ar
> tseries.x.ar.P = tseries.x
> for(i in (o+1):length(tseries.x)){
+ tseries.x.ar.P[i] = sum(rev(a)*tseries.x[(i-o):(i-1)]) + mean(tseries.x)*(1-sum(a))
+ }
```

Rispettivi residui:

```
> res.x.HW = residuals(tseries.x.HW)
> res.x.ar = na.omit(tseries.x.ar$resid)
```

Grafico:

```
> ts.plot(x,predict(tseries.x.HW,12),
+ predict(tseries.x.HW,12) + quantile(res.x.HW,0.05),
+ predict(tseries.x.HW,12) + quantile(res.x.HW,0.95),
+ predict(tseries.x.ar,n.ahead=12,se.fit=FALSE),
+ predict(tseries.x.ar,n.ahead=12,se.fit=FALSE) + quantile(res.x.ar,0.05),
+ predict(tseries.x.ar,n.ahead=12,se.fit=FALSE) + quantile(res.x.ar,0.95),
+ lwd=c(2,1,1,1,1,1,1),
+ col=c("darkgreen","red","blue","blue","darkred","darkblue","darkblue"))
```

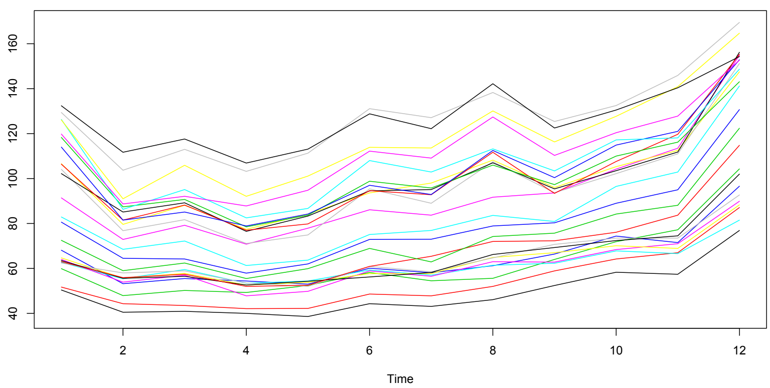


Rieccoci a preferire tutto sommato la seconda, seppur di poco, continuando a riconoscere quanto questa volta siano davvero ottime entrambe, e comunque ribadiamo infine la tendenza della serie ad un'ascesa generale a fronte di eventuali ma leggere discese, magari più istantanee che non, e questo in accordo forse con una certa ciclicità di fondo nella quale tendiamo a credere.

Il problema a questo punto è invece che la serie si arresta all'ormai lontano 2008, per cui potrebbe esser molto più sensato e quindi più utile per i nostri scopi calcolare medie annuali con tanto di banda di confidenza opportunamente ampia: risolviamo in seguito tale elementare questione.

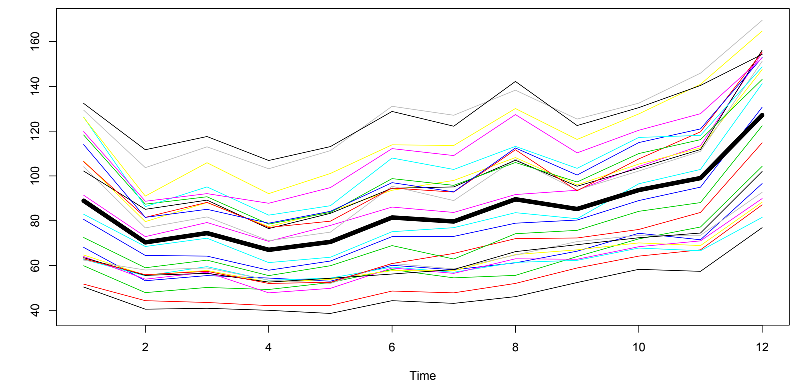
Primissimo plot:

```
> tseries.m = matrix(tseries,12,25)
> ts.plot(ts(tseries.m),col=palette())
```



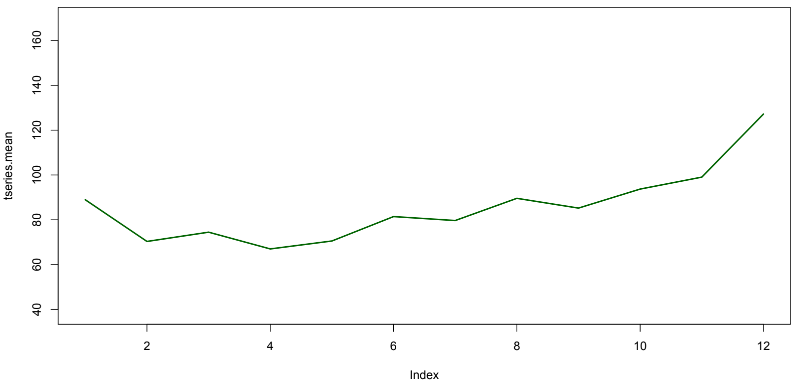
Deduciamone il profilo medio:

```
> tseries.mean = vector("numeric",12)
> for(i in 1:12){
+ tseries.mean[i] = mean(tseries.m[i,])
+ }
> lines(tseries.mean,lwd=6)
```



Isoliamolo da tutto il resto:

```
> plot(tseries.mean,type="l",lwd=2,col="darkgreen",
+ ylim=range(c(min(tseries),max(tseries))))
```

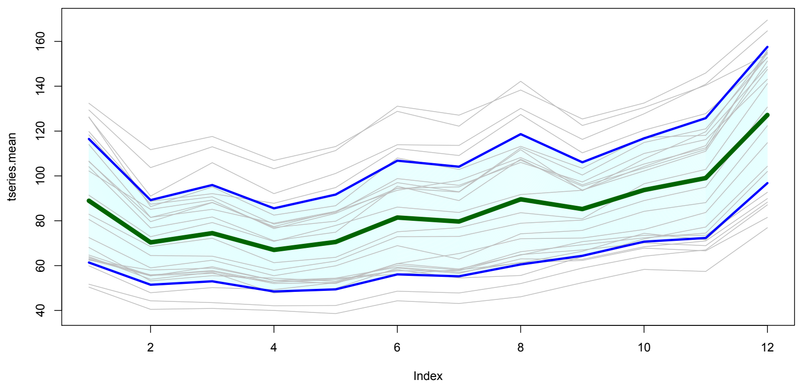


Prepariamoci per disegnarci attorno:

```
> tseries.sd = vector("numeric",12)
> for(i in 1:12){
+ tseries.sd[i] = sd(tseries.m[i,])
+ }
```

Terminiamo:

```
> banda.inf = tseries.mean - tseries.sd
> banda.sup = tseries.mean + tseries.sd
> comb.conv = seq(0,1,0.01)
> for(k in 2:100){
+ lines(comb.conv[k]*banda.sup + (1-comb.conv[k])*banda.inf,col="lightcyan")
+ }
> for(j in 1:25){
+ lines(tseries[1:12 + 12*(j-1)],col="gray")
+ }
> lines(banda.inf,lwd=3,col="blue")
> lines(banda.sup,lwd=3,col="blue")
> lines(tseries.mean,lwd=6,col="darkgreen")
```



Sintesi finale

Le nostre indagini statistiche suggeriscono, in conclusione: un tasso di disoccupazione specialmente giovanile in probabile ciclicità e forse in lieve crescita, ma di certo non in decrescita - coerentemente semmai con la situzione economica di molti paesi europei; dei prezzi delle case in un costante aumento in perfetta continuità con la loro evoluzione negli anni - di nuovo non sorprendente, trattandosi quasi della domanda di case sul mercato; ed infine spese nell'arredamento di interni generici inesorabilmente periodiche, dunque in un certo senso affidabili, ed anzi molto probabilmente in lieve crescita ciclica ma sicuramente per niente in decrescita - ancora niente di strano -, con la riserva di ricordare a riguardo la mancanza di rilevazioni di dati più recenti.

Dunque il nostro rapporto è quasi totalmente positivo e concorre per ciò non tanto ad un immediato termine di tutti i vari ragionamenti per la scelta aziendale che ci ha mosso, quanto piuttosto ad insistere su ulteriori indagini che siano condotte possibilmente su altri dati o perlomeno servendosi di un armamentario tecnico superiore. Detto questo, è del tutto ovvio che per procedere in tale modo occorrerebbe conoscere pure l'esito del lavoro del resto del nostro gruppo, facendo in ogni caso particolare attenzione alla tipologia dei dati risultati meno soddisfacenti e capendo quanto meno siano soddisfacenti rispetto agli altri.

In effetti è proprio l'eventualità di successivi studi per mezzo di strumenti statistici avanzati che ci ha spinto a dare il meglio di noi fin da subito, sebbene ad esempio le nostre serie di dati non fossero delle più imprevedibili, e tutto questo secondo la saggia idea generale di analizzare sempre con la massima cura la situazione pur facendo infine pesare il fatto che per veri guadagni sia sempre necessario un vero rischio.