

```

library(tseries)
library(lattice)
library(graphics)

#as1= c(.033,.043,.051,.059,.061,.063,.053,.036,.046,.056,.063,.048,.053,.043,.066,.053,
#.082,.06,.08,.076,.056,.036,.05,.053,.056,.058,
#.061,.063,.065,.068,.0815,.095,.079,.063,.069,.074,.08,.0765,.073,
#.0695,.066,.093,.083,.073,.063,.074,.067,.06,.086,.08,.073,.067,
#.089,.064,.087,.079,.07,.065,.06,.063)

# Per partire con la detrendizzazione, ad as1 sostituiamo i valori della stessa serie
detrendizzata.

# Togliamo il cancelletto e mettiamo la nuova serie detrendizzata qui e 'cancellettiamo' la
precedente:

as1 =c(-.018,.0089,-.0013,.0062,.0077,.0093,
-.0012,-0.0187,-.0091,.00039,.0069,-.0085,
-.0040,-.014,.0080,-.0054,.0231,.00064,
.0202,.0157,-.0048,-.0252,-.0117,-.0092,
-.0066,-.0051,-.0026,-.0011,.00048,.0030,
.0160,.029,.013,-.0039,.0017,-.0092,
.012,.0076,.0038,-.00018,-.0042,.0223,
.012,.0014,-.0090,.0015,-.0060,-.0134,
.0121,.0056,-.0018,-.0083,.0132,-.00122,
.0102,.0018,-.0077,-.0131,-.0186,-.0161)

par(ask=T)
par(mfrow=c(1,2))
yt=c()
yt=as1
ts.plot(yt, main="GRAF. N.2_yt_ SERIE CORRETTA")
lines(yt,type="l")
acf(yt, main="GRAF. N.2_CORR_SERIE CORRETTA")
#alfa=-pi/2 -> 270°; alfa=-1.175 rad (cioè -100°) -> 260°

```

```
#INIZIO FUNCTION
```

```
PRDGRAM<- function(y1,n1,m1) {
```

```
# VALORI DEL PARAMETRO ak
```

```
aO=c(); k=0; aO=0;
```

```
for(t in 1:n1){aO=aO+y1[t]*cos(2*pi*t*k/n1)}
```

```
aO
```

```
aO=aO*2/n1;aO=aO/2
```

```
aO
```

```
a=c();a[1:m1]=0;
```

```
for(k in 1:m1) {
```

```
for(t in 1:n1){
```

```
a[k]=a[k]+y1[t]*cos(2*pi*t*k/n1)}}}
```

```
a=2*a/n1
```

```
# VALORI DEL PARAMETRO bk
```

```
b=c();b[1:m1]=0;bO=0;k=0
```

```
for(k in 1:m1) {
```

```
for(t in 1:n1){
```

```
b[k]=b[k]+y1[t]*sin(2*pi*t*k/n1)}}}
```

```
a <- as.vector(a)
```

```
for(i in 1:m1){
```

```
if (abs(a[i]) < 1e-10) a[i]=0 else a[i]=a[i]}
```

```
a
```

```
for(i in 1:m1){
```

```
if (abs(b[i]) < 1e-10) b[i]=0 else b[i]=b[i]}
```

```
b=2*b/n1
```

```
b
```

```
# AMPIEZZE
```

```
#ro[1:m1]=0
```

```
ro <- sqrt(a^2 +b^2)
```

```
for(i in 1:m1){
```

```

if (abs(ro[i]) < 1e-10) ro[i]=0 else ro[i]=ro[i]}

# CALCOLO DELLA FASE DI OGNI ARMONICA

# RIPORTANDO IL VALORE AL QUADRANTE GIUSTO

f2=c()

f2[1:m1]=0

for(i in 1:m1){

f2[i] <- abs(a[i]/b[i])

f2[i] <- atan(f2[i])*180/pi}

f2 =as.vector(f2)

f2

#f2[1:m1]=0 un f2[1:m1] di troppo!

phi <- c()

for(i in 1:m1){

# f2 <- abs(a[i]/b[i]);

# f2 <- atan(f2)*180/pi;

if(b[i]>0 & a[i]>0) phi[i] = f2[i];

if(b[i]<0 & a[i]>0) phi[i] = 180-f2[i];

if(b[i]<0 & a[i]<0) phi[i] = 180+f2[i];

if(b[i]>0 & a[i]<0) phi[i] = 360-f2[i];

if(b[i]==0 & a[i]==0) phi[i] = 0;

if((b[i]<0 & b[i]>0) | a[i]==0) phi[i]=0;

if(b[i]==0 & a[i]>0) phi[i]=90;

if(b[i]==0 & a[i]<0) phi[i]=360-90

}

# PHI FASE ARMONICHE

phi=as.vector(phi)

phi

param_a <-a

param_b <-b

ampiezza <- ro

```

```

fase <- phi

# Qui, al termine della function si pone il valore di un'unica
# variabile che esce o, se escono più variabili, si usa
# un data.frame: data=data.frame(x1,x2,...).
# Ogni chiamata alla function permette di includere l'unica
# variabile o i data nel nome della chiamata:
# es. periodxx=nome.function(x1,x2,...)
data <-data.frame(a,b,ro, phi)
data
# questa matrice esce dalla function e viene 'raccolta' nella variabile nomexx (es.,periodxx)
}

FINE FUNCTION

#Per richiamare la function:

#nomexx = PRDGRAM(Nome_var_vettore dati, numerosità del campione,
numero di armoniche da cercare)

yt=as1
yx=as1
nx=length(yt)
#periodogramma yt
if (nx/2== nx%%2) mx=nx/2-1 else mx=(nx-1)/2 #da controllare se non sia necessario
uno swap!
period_as1= PRDGRAM(yx, nx ,mx)

#par(mfrow=c(1,4))
#plot(a, xlab="Armoniche = N° osc. in n dati")
#plot(b, xlab="Armoniche = N° osc. in n dati")

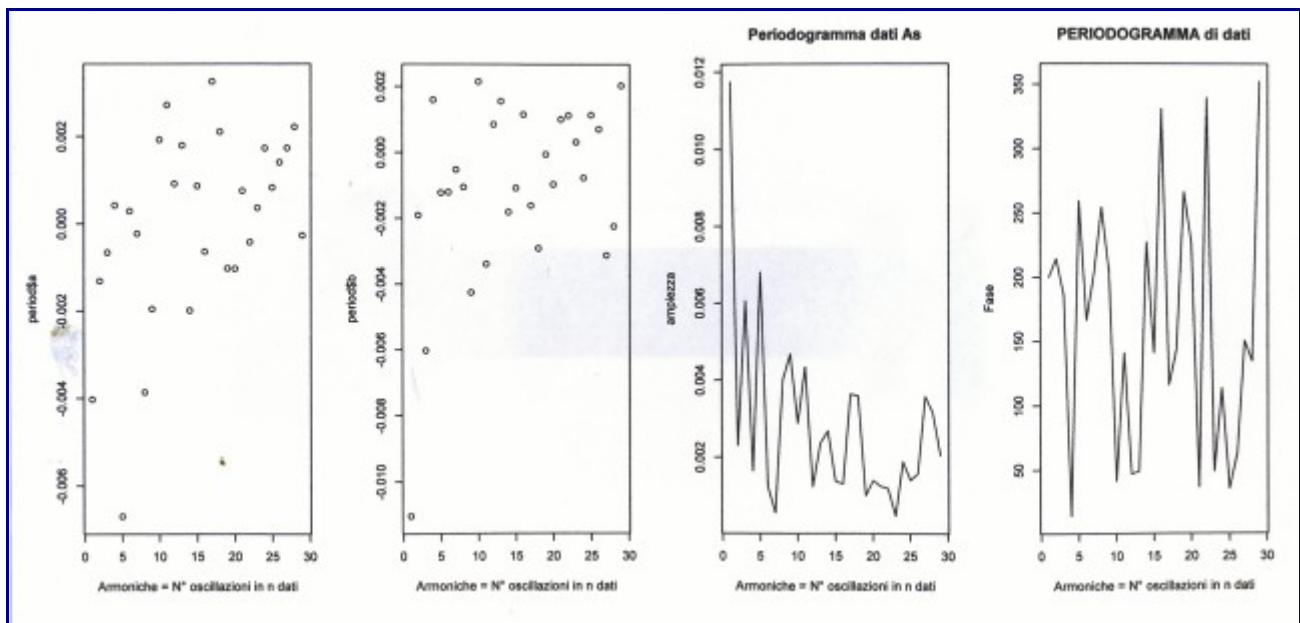
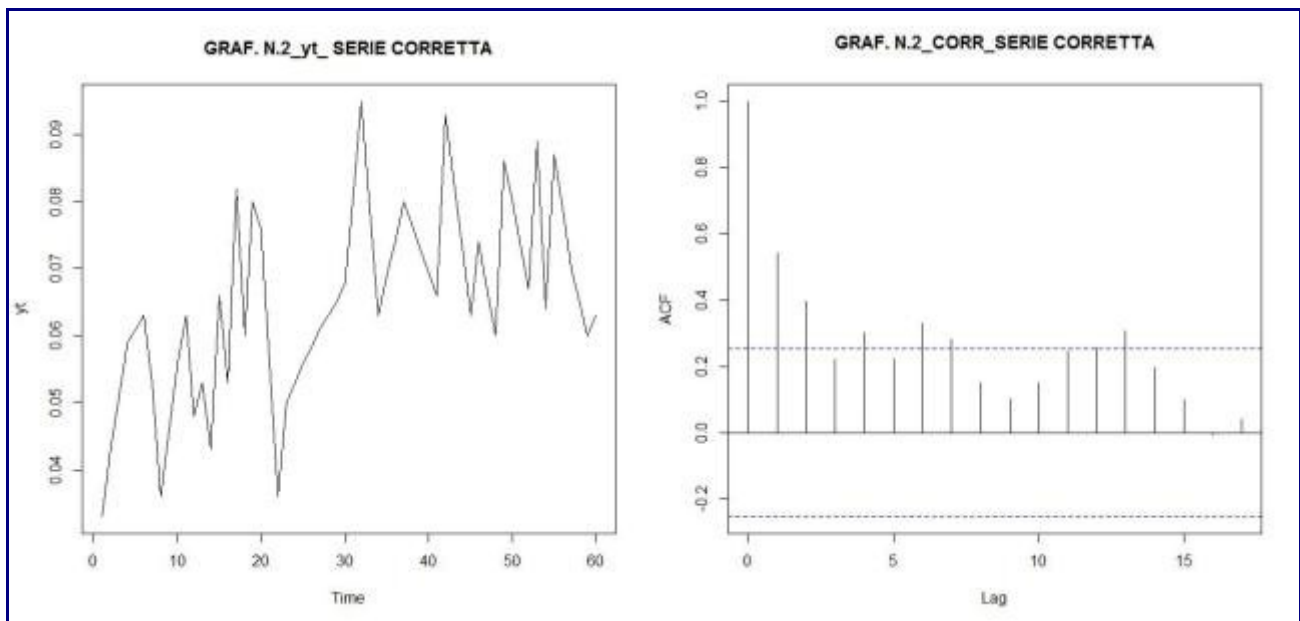
period_as1 # tabella dei dati in uscita: ak, bk, ampiezze, fasi
# Con questa tabella si costruiscono le formule analitiche delle
armoniche

period_as1$ro # vettore delle ampiezze

plot(period_as1$ro,type="l",main="Ampiezze osc di As1",

```

```
xlab="Armoniche = N° oscill. in n dati", ylab="ampiezza")
```



```
par(mfrow=c(1,4))
```

```
plot(period_as1$a,ylab="Parametro a")
```

```
plot(period_as1$b,ylab="Parametro b")
plot(period_as1$ro,type="l",main="PERIODOGRAMMA di as1",
xlab="Armoniche = N° osc. in nx dati", ylab="ampiezza")
plot(period_as1$phi,type="l", ylab="Fase")
```

#Per vedere i risultati trasferiti dalla consolle di R in pdf
#del precedente frammento di programma cliccare sotto:

As1_corr_R

```
par(mfrow=c(1,1))
```

```
as1.ts1=ts(as1,start=1989,frequency=12)
subas=as1.ts1[seq(1,length(as1),by=12)]
```

```
#-----
```

Gli scripts che riguardano il calcolo delle variabili vettoriali mediamesi e Mmesio per ora sono esclusi.

```
#mediamesi=c()
```

```
#for(i in 1:12){mediamesi[i]=mean(as1.ts1[seq(i,length(as1),by=12)])}
```

```
#ts.plot(mediamesi,main"mediamesi in 5 anni")
```

```
#Mmesio=c()
```

```
#a=mediamesi
```

```
#b=mean(as1)
```

```
#c=a-b
```

#Mmesio=c () 12 valori medi meno la media serie originale; una specie di Effetto Stagionale

```
#Mmesio=mediamesi – mean(as1)
```

```
#ts.plot(Mmesio) # da controllare: Effetto Stagionale da confrontare con mediacol
```

```
#acf(Mmesio, main="CORR_Mmesio")
```

```
#Mmesio # da confrontare con mediacol
```

```
#-----
```

```
yt=as1
```

```
yt=as.vector(yt); n=length(yt); Mbt=c()
```

```
for(t in 7:n){Mbt[t] = (yt[t-6]/2+yt[t-5]+yt[t-4]+yt[t-3]+yt[t-2]+
```

```

yt[t-1]+yt[t]+yt[t+1]+yt[t+2]+yt[t+3]+yt[t+4]+yt[t+5]+(yt[t+6])/2)/12}
Mbt #è quello che resta di as1, dopo la media mobile 12 (trend-ciclo_random
Mbt=Mbt[7:54]# elimino da Mbt gli NA
ts.plot(Mbt, main="GRAF. N.4-Serie as1 destag.")
acf(Mbt, main="GRAF. N.4_Serie as1 destag.")
#Periodogramma Mbt, serie più corta senza stagionalità
y3=c()
y3=Mbt
n3=length(y3)
if (n3/2== n3%%2) m3=n3/2-1 else m3=(n3-1)/2
#ifelse(nx%%2 > 0, m=(n-1)/2, m=n/2-1
period_Mbt=PRDGRAM(y3, n3 ,m3)
period_Mbt # tabella ak, bk,ro,phi
period_Mbt$ro #valori ampiezza di Mbt
ts.plot(period_Mbt$ro, main="GRAF. N.4-Serie destag.")

# Filtro col comando filter la serie yt
asf12=filter(yt, filter=rep(1/13,13))
asf12
asf12=asf12[7:54] # elimino da asf12 gli NA
#Mbt contiene l'as1 senza la stagionalità; in as1 però rimane quello
#che aveva ( trend-stagionalità-ciclo_random); se da as1, tolgo as1 senza la stagionalità,
#trovo la stagionalità e random (STRD) che trasformo in Effetto Stagionale eliminando
#una buona parte dei random.

STRD=as1[7:54]-Mbt # componente stagionale + random, serie più corta
STRD # da essa si estraggono gli Effetti Stagionali
#Processo per costruire gli Effetti Sagionali attraverso STRD
stag = matrix(STRD, ncol=12, byrow=T)
mediacol = colMeans(stag) #in mediacol rimangono i random? o si perdono nella

```

mediazione.

in questo primo mediacol ottengo 12 valori a partire da luglio.

mediacol=c(mediacol[7:12], mediacol[1:6]) # qui ordino da gennaio a dicembre i 12

#valori dell' EFFETTO STAGIONALE

mediacol # è detto anche Fattore Stagionale

#ts.plot(mediacol)

ESAs = rep(mediacol,5) # 'copro' con l'Effetto Stagionale di yt o as1 i 60 dati

ESAs #serie lunga come yt o as1 originale

ts.plot(ESAs,main="GRAF. N.3-EFFETTO STAGIONALE")

ESAs1 = rep(mediacol,2)

ts.plot(ESAs1,main="GRAF. N.3-EFFETTO STAGIONALE")

acf(ESAs1, main="GRAF. N.3 -CORR-EF. STAG. 2 ripet")

#periodogramma ESAs1

yes=ESAs1

nes=length(ESAs1)

if (nes/2== nes%%2) mes=nes/2-1 else mes=(nes-1)/2

period_ESAs1=PRDGRAM(yes, nes, mes)

period_ESAs1

period_ESAs1\$ro

plot(period_ESAs1\$ro,type="l", main="Period_ro EFFETTO STAG.")

period_ESAs1

period_ESAs1\$ro

plot(period_ESAs1\$ro,type="l",main="Period-stag.")

dst=c() #attivo la serie destagionalizzata

dst=as1-ESAs1# da provare se funziona

dst

#e=as1

#f=ESAs

#g=e-f

```

#dst=g

#Potrei smussare dst con una Media Mobile Pesata (3*3, cioè con pesi 1,2,3,2,1) per
tentare di eliminare la componente casuale

#Si otterrebbe una serie (y1t) contenente CICLO+TREND, che se la tolgo dalla serie
destagionalizzata dst precedente dovrei ottenere il #RESIDUO.

yd=dst
nd=length(dst)
if (nd/2== nd%%2) md=nd/2-1 else md=(nd-1)/2
period_dst=PRDGRAM(yd, nd, md)
period_dst
period_dst$ro
plot(period_dst$ro,type="l",main="Period.destag. GRAF.N.4, b")


#PROVIAMO INVECE A TOGLIERE IL TREND DALLA dst

plot(dst,type="l", main="yt-destagionalizzata") # la y1t o dst: yt destagionalizzata=
ciclo+TREND +random (GRAF. N.4, a), anche ottenuti col programma Excel)
acf(dst, main="dst-CORR-yt-destag")

# da fare il periodogramma di dst ,(GRAF. N.4, b)

# Se elimino il TREND da dst ottengo CLRD e posso controllare con CORR se
# ciò che resta è da considerare residuo. yt-ESAs-TREND = CLRD
# CLRD =yt-TREND- ESAs
#Calcolo il trend di dst per toglierlo da yt-ESAs ed ottenere CLRD
t=seq(1:60)
fitdst=lm(dst~t)
abline(lm(dst~t))
summary(fitdst)
resid(fitdst)
p=predict(fitdst,data.frame(t=c(1,60)))
CLRD=c()
CLRD=(dst-p)

```

```

CLRD
CLRD=yt-ESAs-p
n1=length(p)
ts.plot(CLRD, main="GRAF. N.5-RESIDUI" )
acf(CLRD, main="GRAF. N.5-CORR_RESIDUI")
#periodogramma di CLRD
yr=CLRD
nr=length(yx)
if (nr/2== nr%%2) mr=nr/2-1 else mr=(nr-1)/2
period_clrd=PRDGRAM(yr, nr ,mr)
period_clrd
period_clrd$ro
plot(period_clrd$ro,type="l",main="Period.ciclo_random")
#da controllare ancora

```

```
> rm(list=ls(all=TRUE))
```

```
> PARTE IN SINTESI è stato controllato.
```

```
Errore: unexpected symbol in "PARTE IN"
```

```
>
```

```
> #DA QUI SI INIZIA COPIARE IL PROGRAMMA IN BLOCCO NOTE (in via di controllo)
```

```
>
```

```
> library(tseries)
```

```
‘tseries’ version: 0.10-32
```

‘tseries’ is a package for time series analysis and computational finance.

See ‘library(help="tseries")’ for details.

```

>
> library(lattice)
>
> library(graphics)
>
> #as1=
c(.033,.043,.051,.059,.061,.063,.053,.036,.046,.056,.063,.048,.053,.043,.066
,.053,
>
> #.082,.06,.08,.076,.056,.036,.05,.053,.056,.058,
>
> #.061,.063,.065,.068,.0815,.095,.079,.063,.069,.074,.08,.0765,.073,
>
> #.0695,.066,.093,.083,.073,.063,.074,.067,.06,.086,.08,.073,.067,
>
> #.089,.064,.087,.079,.07,.065,.06,.063)
>
> # Per partire con la detrendizzazione, ad as1 sostituiamo i valori della stessa
serie detrendizzata.
>
> # Togliamo il cancelletto e mettiamo la nuova serie detrendizzata qui e
'cancelliamo' la precedente:
>
> as1 = c(-.018,.0089,-.0013,.0062,.0077,.0093,
+
+ -.0012,-0.0187,-.0091,.00039,.0069,-.0085,
+
+ -.0040,-.014,.0080,-.0054,.0231,.00064,
+
+ .0202,.0157,-.0048,-.0252,-.0117,-.0092,

```

```

+
+ -.0066,-.0051,-.0026,-.0011,.00048,.0030,
+
+ .0160,.029,.013,-.0039,.0017,-.0092,
+
+ .012,.0076,.0038,-.00018,-.0042,.0223,
+
+ .012,.0014,-.0090,.0015,-.0060,-.0134,
+
+ .0121,.0056,-.0018,-.0083,.0132,-.00122,
+
+ .0102,.0018,-.0077,-.0131,-.0186,-.0161)
>
> par(ask=T)
>
> par(mfrow=c(1,2))
>
> yt=c()
>
> yt=as1
>
> ts.plot(yt, main="GRAF. N.2_yt_ SERIE CORRETTA")
Aspetto per confermare cambio pagina...
>
> lines(yt,type="l")
>
> acf(yt, main="GRAF. N.2_CORR_SERIE CORRETTA")
>
> #alfa=-pi/2 -> 270°; alfa=-1.175 rad (cioè -100°) -> 260°
>

```

```

>
>
> #INIZIO FUNCTION
>
> PRDGRAM<- function(y1,n1,m1) {
+
+ # VALORI DEL PARAMETRO ak
+
+ a0=c(); k=0; a0=0;
+
+ for(t in 1:n1){a0=a0+y1[t]*cos(2*pi*t*k/n1)}
+
+ a0
+
+
+ a0=a0*2/n1;a0=a0/2
+
+ a0
+
+
+ a=c();a[1:m1]=0;
+
+ for(k in 1:m1) {
+
+ for(t in 1:n1){
+
+ a[k]=a[k]+y1[t]*cos(2*pi*t*k/n1)}}
+
+ a=2*a/n1
+
+ # vALORI DEL PARAMETRO bk
+

```

```

+ b=c();b[1:m1]=0;bo=0;k=0
+
+ for(k in 1:m1) {
+
+ for(t in 1:n1){
+
+ b[k]=b[k]+y1[t]*sin(2*pi*t*k/n1)}}
+
+ a <- as.vector(a)
+
+ for(i in 1:m1){
+
+ if (abs(a[i]) < 1e-10) a[i]=0 else a[i]=a[i]}
+
+ a
+
+ for(i in 1:m1){
+
+ if (abs(b[i]) < 1e-10) b[i]=0 else b[i]=b[i]}
+
+ b=2*b/n1
+
+ b
+
+ # AMPIEZZE
+
+ #ro[1:m1]=0
+
+ ro <- sqrt(a^2 +b^2)
+

```

```

+ for(i in 1:m1){
+
+ if (abs(ro[i]) < 1e-10) ro[i]=0 else ro[i]=ro[i]}
+
+ # CALCOLO DELLA FASE DI OGNI ARMONICA
+
+ # RIPORTANDO IL VALORE AL QUADRANTE GIUSTO
+
+ f2=c()
+
+ f2[1:m1]=0
+
+ for(i in 1:m1){
+
+ f2[i] <- abs(a[i]/b[i])
+
+ f2[i] <- atan(f2[i])*180/pi}
+
+ f2 =as.vector(f2)
+
+ f2
+
+ #f2[1:m1]=0 un f2[1:m1] di troppo!
+
+ phi <- c()
+
+ for(i in 1:m1){
+
+ # f2 <- abs(a[i]/b[i]);
+

```

```

+ # f2 <- atan(f2)*180/pi;
+
+ if(b[i]>0 & a[i]>0) phi[i] = f2[i];
+
+ if(b[i]<0 & a[i]>0) phi[i] = 180-f2[i];
+
+ if(b[i]<0 & a[i]<0) phi[i] = 180+f2[i];
+
+ if(b[i]>0 & a[i]<0) phi[i] = 360-f2[i];
+
+ if(b[i]==0 & a[i]==0) phi[i] = 0;
+
+ if((b[i]<0 & b[i]>0) | a[i]==0) phi[i]=0;
+
+ if(b[i]==0 & a[i]>0) phi[i]=90;
+
+ if(b[i]==0 & a[i]<0) phi[i]=360-90
+
+ }
+
+ # PHI FASE ARMONICHE
+
+ phi=as.vector(phi)
+
+ phi
+
+ param_a <-a
+
+ param_b <-b
+

```

```

+ ampiezza <- ro
+
+ fase <- phi
+
+ # Qui, al termine della function si pone il valore di un'unica
+
+ # variabile che esce o, se escono più variabili, si usa
+
+ # un data.frame: data=data.frame(x1,x2,...).
+
+ # Ogni chiamata alla function permette di includere l'unica
+
+ # variabile o i data nel nome della chiamata:
+
+ # es. periodxx=nome.function(x1,x2,...)
+
+ data <-data.frame(a,b,ro, phi)
+
+ data
+
+ # questa matrice esce dalla function e viene 'raccolta' nella variabile nomexx
(es.,periodxx)
+
+ }
>
> FINE FUNCTION
Errore: unexpected symbol in "FINE FUNCTION"
>
> #Per richiamare la function:
>

```

```

> #nomexx = PRDGRAM(Nome_var_vettore_dati, numerosità del campione,
numero di armoniche da cercare)
>
> yt=as1
>
> yx=as1
>
> nx=length(yt)
>
> #periodogramma yt
>
> if (nx/2== nx%%2) mx=nx/2-1 else mx=(nx-1)/2 #da controllare se non sia
necessario uno swap!
>
> period_as1= PRDGRAM(yx, nx ,mx)
>
> #par(mfrow=c(1,4))
> #plot(a, xlab="Armoniche = N° osc. in n dati")
> #plot(b, xlab="Armoniche = N° osc. in n dati")
>
> period_as1 # tabella dei dati in uscita: ak, bk, ampiezze, fasi
      a      b      ro      phi
1 -3.204012e-03 -1.866197e-03 0.0037078813 239.781075
2 -1.284547e-03 1.975535e-03 0.0023564379 326.966987
3 -9.217332e-04 -2.605557e-03 0.0027637869 199.481576
4 4.871699e-04 3.740973e-03 0.0037725603 7.419616
5 -7.719706e-03 1.048531e-03 0.0077905888 277.734876
6 1.066929e-04 1.341238e-03 0.0013454750 4.548191
7 -8.956983e-04 1.147146e-03 0.0014554103 322.017071
8 -4.418815e-03 1.419430e-03 0.0046411966 287.808267
9 -1.869250e-03 -2.845106e-03 0.0034042215 213.305138

```

10 1.006833e-03 3.504805e-03 0.0036465559 16.027892
 11 2.561849e-03 -2.137628e-03 0.0033365439 129.841895
 12 -8.629213e-05 9.949849e-04 0.0009987198 355.043307
 13 5.361577e-04 2.532531e-03 0.0025886635 11.953489
 14 -2.923822e-03 -1.665122e-03 0.0033647236 240.338400
 15 -1.089667e-03 -5.806667e-04 0.0012347256 241.947530
 16 -1.569426e-03 1.940568e-03 0.0024957766 321.035940
 17 1.970302e-03 -1.584214e-03 0.0025282054 128.800888
 18 1.144974e-03 -2.073161e-03 0.0023683246 151.088829
 19 -1.204297e-03 -2.636146e-04 0.0012328117 257.652985
 20 -1.935833e-03 -1.188476e-03 0.0022715468 238.452786
 21 8.279972e-04 7.563287e-04 0.0011214332 47.590063
 22 -9.924779e-04 -7.026236e-05 0.0009949619 265.950508
 23 -3.479709e-04 7.134998e-05 0.0003552106 281.587646
 24 1.545292e-03 -1.692149e-03 0.0022915706 137.597270
 25 -2.326275e-04 7.408028e-04 0.0007764692 342.566625
 26 1.468179e-03 9.172080e-04 0.0017311329 58.005960
 27 1.457653e-03 -3.535887e-03 0.0038245588 157.596319
 28 2.211071e-03 -1.569349e-03 0.0027114003 125.365937
 29 5.810030e-04 1.812466e-03 0.0019033121 17.773741

> # Con questa tabella si costruiscono le formule analitiche delle armoniche
>

> period_as1\$ro # vettore delle ampiezze

[1] 0.0037078813 0.0023564379 0.0027637869 0.0037725603
0.0077905888

[6] 0.0013454750 0.0014554103 0.0046411966 0.0034042215
0.0036465559

[11] 0.0033365439 0.0009987198 0.0025886635 0.0033647236
0.0012347256

[16] 0.0024957766 0.0025282054 0.0023683246 0.0012328117
0.0022715468

```
[21] 0.0011214332 0.0009949619 0.0003552106 0.0022915706  
0.0007764692
```

```
[26] 0.0017311329 0.0038245588 0.0027114003 0.0019033121
```

```
>
```

```
> plot(period_as1$ro,type="l",main="Ampiezze osc di As1",  
+ xlab="Armoniche = N° oscill. in n dati", ylab="ampiezza")
```

Aspetto per confermare cambio pagina...

```
>
```

```
>
```

```
>
```

```
> par(mfrow=c(1,4))
```

```
>
```

```
> plot(period_as1$a,ylab="Parametro a")
```

Aspetto per confermare cambio pagina...

```
> plot(period_as1$b,ylab="Parametro b")
```

```
> plot(period_as1$ro,type="l",main="PERIODOGRAMMA di as1",  
+ xlab="Armoniche = N° osc. in nx dati", ylab="ampiezza")
```

```
> plot(period_as1$phi,type="l", ylab="Fase")
```

```
>
```

```
> #Per vedere i risultati trasferiti dalla consolle di R in pdf
```

```
> #del precedente frammento di programma cliccare sotto:
```

```
> par(mfrow=c(1,1))
```

```
>
```

```
> as1.ts1=ts(as1,start=1989,frequency=12)
```

```
> subas=as1.ts1[seq(1,length(as1),by=12)]
```

```
>
```

```
>
```

```
> mediamesi=c()
```

```
>
```

```
>
```

```

>
> for(i in 1:12){mediamesi[i]=mean(as1.ts1[seq(i,length(as1),by=12)])}
>
>
>
> #ts.plot(mediamesi,main"mediamesi in 5 anni")

>
>
>
> #Mmesio=c()
>
> #a=mediamesi
>
> #b=mean(as1)
>
> #c=a-b
>
> #Mmesio=c () 12 valori medi meno la media serie originale; una specie di
Effetto Stagionale
>
> #Mmesio=mediamesi – mean(as1)
>
> #ts.plot(Mmesio) # da controllare: Effetto Stagionale da confrontare con
mediacol
>
> #acf(Mmesio, main="CORR_Mmesio")
>
> #Mmesio # da confrontare con mediacol
>
> yt=as1

```

```

>
> yt=as.vector(yt); n=length(yt); Mbt=c()
>
> for(t in 7:n){Mbt[t] = (yt[t-6]/2+yt[t-5]+yt[t-4]+yt[t-3]+yt[t-2]+
+
+ yt[t-1]+yt[t]+yt[t+1]+yt[t+2]+yt[t+3]+yt[t+4]+yt[t+5]+(yt[t+6])/2)/12}
>
> Mbt #è quello che resta di as1, dopo la media mobile 12 (trend-
ciclo_random
[1]      NA      NA      NA      NA      NA
[6]      NA -0.0008675000 -0.0012383333 -0.0018050000
-0.0019008333
[11] -0.0017425000 -0.0014616667 -0.0009308333 0.0013941667
0.0030066667
[16] 0.0021195833 0.0002783333 -0.0005258333 -0.0006633333
-0.0004008333
[21] -0.0004716667 -0.0007341667 -0.0014975000 -0.0023416667
-0.0024183333
[26] -0.0020391667 -0.0007433333 0.0008858333 0.0023316667
0.0028900000
[31] 0.0036650000 0.0049691667 0.0057650000 0.0060700000
0.0059133333
[36] 0.0065225000 0.0071600000 0.0058433333 0.0037766667
0.0030850000
[41] 0.0029891667 0.0024933333 0.0023225000 0.0022433333
0.0019266667
[46] 0.0013550000 0.0017416667 0.0014866667 0.0004316667
0.0003733333
[51] 0.0004441667 -0.0001100000 -0.0012433333 -0.0018808333
NA
[56]      NA      NA      NA      NA      NA
>
> Mbt=Mbt[7:54]# elimino da Mbt gli NA

```

```

>
> ts.plot(Mbt, main="GRAF. N.4-Serie as1 destag.")
Aspetto per confermare cambio pagina...
>
> acf(Mbt, main="GRAF. N.4_Serie as1 destag.")
Aspetto per confermare cambio pagina...
>
> #Periodogramma Mbt, serie più corta senza stagionalità
>
> y3=c()
>
> y3=Mbt
>
> n3=length(y3)
>
> if (n3/2== n3%%2) m3=n3/2-1 else m3=(n3-1)/2
>
> #ifelse(nx%%2 > 0, m=(n-1)/2, m=n/2-1
>
> period_Mbt=PRDGRAM(y3, n3 ,m3)

> period_Mbt # tabella ak, bk,ro,phi
>
> period_Mbt$ro #valori ampiezza di Mbt

>
> ts.plot(period_Mbt$ro, main="GRAF. N.4-Serie destag.")
>
>
>

```

```

> # Filtro col comando filter la serie yt
>
> asf12=filter(yt, filter=rep(1/13,13))
>
> asf12
Time Series:
Start = 1
End = 60
Frequency = 1
[1]      NA      NA      NA      NA      NA
[6]      NA -1.646923e-03 -1.339231e-03 -1.408462e-03 -1.723846e-03
[11] -4.238462e-04 -9.669231e-04 -1.284615e-04 1.171538e-03 2.240769e-
03
[16] 1.002308e-03 7.230769e-05 -1.166154e-03 -1.020000e-03 -1.104615e-
03
[21] -2.276923e-04 -9.276923e-04 -4.753846e-04 -2.021538e-03
-8.400000e-04
[26] -1.630769e-04 -3.707692e-04 -3.015385e-04 1.767692e-03 1.960000e-
03
[31] 3.590769e-03 4.683077e-03 5.367692e-03 5.553846e-03 5.315385e-
03
[36] 6.993846e-03 7.686154e-03 6.563077e-03 3.640000e-03 2.755385e-
03
[41] 2.593846e-03 1.432308e-03 3.070769e-03 2.578462e-03 1.855385e-
03
[46] 9.246154e-04 1.953846e-03 2.183077e-03 1.252308e-03 4.676923e-
04
[51] -2.323077e-04 -5.476923e-04 -2.093846e-03 -2.870769e-03      NA
[56]      NA      NA      NA      NA      NA
>
> asf12=asf12[7:54] # elimino da asf12 gli NA
>

```

```

> #Mbt contiene l'as1 senza la stagionalità; in as1 però rimane quello
> #che aveva ( trend-stagionalità-ciclo_random); se da as1, tolgo as1 senza la
stagionalità,
> #trovo la stagionalità e random (STRD) che trasformo in Effetto Stagionale
eliminando
> #una buona parte dei random.
>
>
>
> STRD=as1[7:54]-Mbt # componente stagionale + random, serie più corta
>
> STRD # da essa si estraggono gli Effetti Stagionali
[1] -3.325000e-04 -1.746167e-02 -7.295000e-03 2.290833e-03 8.642500e-
03
[6] -7.038333e-03 -3.069167e-03 -1.539417e-02 4.993333e-03 -7.519583e-
03
[11] 2.282167e-02 1.165833e-03 2.086333e-02 1.610083e-02 -4.328333e-
03
[16] -2.446583e-02 -1.020250e-02 -6.858333e-03 -4.181667e-03
-3.060833e-03
[21] -1.856667e-03 -1.985833e-03 -1.851667e-03 1.100000e-04 1.233500e-
02
[26] 2.403083e-02 7.235000e-03 -9.970000e-03 -4.213333e-03
-1.572250e-02
[31] 4.840000e-03 1.756667e-03 2.333333e-05 -3.265000e-03 -7.189167e-
03
[36] 1.980667e-02 9.677500e-03 -8.433333e-04 -1.092667e-02 1.450000e-
04
[41] -7.741667e-03 -1.488667e-02 1.166833e-02 5.226667e-03 -2.244167e-
03
[46] -8.190000e-03 1.444333e-02 6.608333e-04
>
> #Processo per costruire gli Effetti Sagionali attraverso STRD

```

```

>
> stag = matrix(STRD, ncol=12, byrow=T)
>
> mediacol = colMeans(stag) #in mediacol rimangono i random? o si perdono
nella mediazione.
>
> # in questo primo mediacol ottengo 12 valori a partire da luglio.
>
> mediacol=c(mediacol[7:12], mediacol[1:6]) # qui ordino da gennaio a
dicembre i 12
>
> #valori dell' EFFETTO STAGIONALE
>
> mediacol # è detto anche Fattore Stagionale
[1] 0.0023143750 -0.0028679167 0.0002289583 -0.0052401042
0.0070560417
[6] 0.0054358333 0.0106358333 0.0054566667 -0.0038287500
-0.0080000000
[11] -0.0033787500 -0.0111264583
>
> #ts.plot(mediacol)
>
> ESAs = rep(mediacol,5) # 'copro' con l'Effetto Stagionale di yt o as1 i 60 dati
>
> ESAs #serie lunga come yt o as1 originale
[1] 0.0023143750 -0.0028679167 0.0002289583 -0.0052401042
0.0070560417
[6] 0.0054358333 0.0106358333 0.0054566667 -0.0038287500
-0.0080000000
[11] -0.0033787500 -0.0111264583 0.0023143750 -0.0028679167
0.0002289583
[16] -0.0052401042 0.0070560417 0.0054358333 0.0106358333

```

0.0054566667

[21] -0.0038287500 -0.0080000000 -0.0033787500 -0.0111264583
0.0023143750

[26] -0.0028679167 0.0002289583 -0.0052401042 0.0070560417
0.0054358333

[31] 0.0106358333 0.0054566667 -0.0038287500 -0.0080000000
-0.0033787500

[36] -0.0111264583 0.0023143750 -0.0028679167 0.0002289583
-0.0052401042

[41] 0.0070560417 0.0054358333 0.0106358333 0.0054566667
-0.0038287500

[46] -0.0080000000 -0.0033787500 -0.0111264583 0.0023143750
-0.0028679167

[51] 0.0002289583 -0.0052401042 0.0070560417 0.0054358333
0.0106358333

[56] 0.0054566667 -0.0038287500 -0.0080000000 -0.0033787500
-0.0111264583

>

> ts.plot(ESAs,main="GRAF. N.3-EFFETTO STAGIONALE")

Aspetto per confermare cambio pagina...

>

>

>

> ESAs1 = rep(mediacol,2)

>

> ts.plot(ESAs1,main="GRAF. N.3-EFFETTO STAGIONALE")

Aspetto per confermare cambio pagina...

>

> acf(ESAs1, main="GRAF. N.3 -CORR-EF. STAG. 2 ripet")

Aspetto per confermare cambio pagina...

>

> #periodogramma ESAs1

```

>
> yes=ESAs1
>
> nes=length(ESAs1)
>
> if (nes/2== nes%%2) mes=nes/2-1 else mes=(nes-1)/2
>
> period_ESAs1=PRDGRAM(yes, nes, mes)
>
> period_ESAs1
      a      b      ro      phi
1  0.0000000000 0.000000e+00 0.0000000000 0.000000
2 -0.0063913200 4.920238e-05 0.0063915094 270.44107
3  0.0000000000 0.000000e+00 0.0000000000 0.000000
4  0.0019247656 3.623129e-03 0.0041026556 27.97921
5  0.0000000000 0.000000e+00 0.0000000000 0.000000
6 -0.0009129688 -3.240625e-04 0.0009687768 250.45745
7  0.0000000000 0.000000e+00 0.0000000000 0.000000
8 -0.0020464149 -9.462681e-04 0.0022546036 245.18404
9  0.0000000000 0.000000e+00 0.0000000000 0.000000
10 -0.0009768570 1.655589e-03 0.0019222970 329.45787
11 0.0000000000 0.000000e+00 0.0000000000 0.000000
>
> period_ESAs1$ro
[1] 0.0000000000 0.0063915094 0.0000000000 0.0041026556
0.0000000000
[6] 0.0009687768 0.0000000000 0.0022546036 0.0000000000
0.0019222970
[11] 0.0000000000
>
> plot(period_ESAs1$ro,type="l", main="Period_ro EFFETTO STAG.")

```

Aspetto per confermare cambio pagina...

>

> period_ESAs1

a b ro phi

```
1 0.0000000000 0.000000e+00 0.0000000000 0.000000
2 -0.0063913200 4.920238e-05 0.0063915094 270.44107
3 0.0000000000 0.000000e+00 0.0000000000 0.000000
4 0.0019247656 3.623129e-03 0.0041026556 27.97921
5 0.0000000000 0.000000e+00 0.0000000000 0.000000
6 -0.0009129688 -3.240625e-04 0.0009687768 250.45745
7 0.0000000000 0.000000e+00 0.0000000000 0.000000
8 -0.0020464149 -9.462681e-04 0.0022546036 245.18404
9 0.0000000000 0.000000e+00 0.0000000000 0.000000
10 -0.0009768570 1.655589e-03 0.0019222970 329.45787
11 0.0000000000 0.000000e+00 0.0000000000 0.000000
```

>

> period_ESAs1\$ro

```
[1] 0.0000000000 0.0063915094 0.0000000000 0.0041026556
0.0000000000
```

```
[6] 0.0009687768 0.0000000000 0.0022546036 0.0000000000
0.0019222970
```

```
[11] 0.0000000000
```

>

> plot(period_ESAs1\$ro,type="l",main="Period-stag.")

Aspetto per confermare cambio pagina...

>

> dst=c() #attivo la serie destagionalizzata

>

> dst=as1-ESAs1# da provare se funziona

Warning message:

In as1 - ESAs1 :

longer object length is not a multiple of shorter object length

>

> dst

[1] -0.0203143750 0.0117679167 -0.0015289583 0.0114401042
0.0006439583

[6] 0.0038641667 -0.0118358333 -0.0241566667 -0.0052712500
0.0083900000

[11] 0.0102787500 0.0026264583 -0.0063143750 -0.0111320833
0.0077710417

[16] -0.0001598958 0.0160439583 -0.0047958333 0.0095641667
0.0102433333

[21] -0.0009712500 -0.0172000000 -0.0083212500 0.0019264583
-0.0089143750

[26] -0.0022320833 -0.0028289583 0.0041401042 -0.0065760417
-0.0024358333

[31] 0.0053641667 0.0235433333 0.0168287500 0.0041000000
0.0050787500

[36] 0.0019264583 0.0096856250 0.0104679167 0.0035710417
0.0050601042

[41] -0.0112560417 0.0168641667 0.0013641667 -0.0040566667
-0.0051712500

[46] 0.0095000000 -0.0026212500 -0.0022735417 0.0097856250
0.0084679167

[51] -0.0020289583 -0.0030598958 0.0061439583 -0.0066558333
-0.0004358333

[56] -0.0036566667 -0.0038712500 -0.0051000000 -0.0152212500
-0.0049735417

>

> #e=as1

>

> #f=ESAs

>

> #g=e-f

```

>
> #dst=g
>
> #Potrei smussare dst con una Media Mobile Pesata (3*3, cioè con pesi
1,2,3,2,1) per tentare di eliminare la componente casuale
>
> #Si otterrebbe una serie (y1t) contenente CICLO+TREND, che se la tolgo
dalla serie destagionalizzata dst precedente dovrei ottenere il #RESIDUO.
>
> yd=dst
>
> nd=length(dst)
>
> if (nd/2== nd%%2) md=nd/2-1 else md=(nd-1)/2
>
> period_dst=PRDGRAM(yd, nd, md)
>
> period_dst
      a      b      ro      phi
1 -3.204012e-03 -1.866197e-03 0.0037078813 239.781075
2 -1.284547e-03 1.975535e-03 0.0023564379 326.966987
3 -9.217332e-04 -2.605557e-03 0.0027637869 199.481576
4 4.871699e-04 3.740973e-03 0.0037725603 7.419616
5 -1.328386e-03 9.993281e-04 0.0016623073 306.953690
6 1.066929e-04 1.341238e-03 0.0013454750 4.548191
7 -8.956983e-04 1.147146e-03 0.0014554103 322.017071
8 -4.418815e-03 1.419430e-03 0.0046411966 287.808267
9 -1.869250e-03 -2.845106e-03 0.0034042215 213.305138
10 -9.179323e-04 -1.183237e-04 0.0009255270 262.654936
11 2.561849e-03 -2.137628e-03 0.0033365439 129.841895
12 -8.629213e-05 9.949849e-04 0.0009987198 355.043307

```

13 5.361577e-04 2.532531e-03 0.0025886635 11.953489
14 -2.923822e-03 -1.665122e-03 0.0033647236 240.338400
15 -1.766979e-04 -2.566042e-04 0.0003115571 214.551342
16 -1.569426e-03 1.940568e-03 0.0024957766 321.035940
17 1.970302e-03 -1.584214e-03 0.0025282054 128.800888
18 1.144974e-03 -2.073161e-03 0.0023683246 151.088829
19 -1.204297e-03 -2.636146e-04 0.0012328117 257.652985
20 1.105816e-04 -2.422075e-04 0.0002662569 155.460583
21 8.279972e-04 7.563287e-04 0.0011214332 47.590063
22 -9.924779e-04 -7.026236e-05 0.0009949619 265.950508
23 -3.479709e-04 7.134998e-05 0.0003552106 281.587646
24 1.545292e-03 -1.692149e-03 0.0022915706 137.597270
25 7.442295e-04 -9.147865e-04 0.0011792845 140.869734
26 1.468179e-03 9.172080e-04 0.0017311329 58.005960
27 1.457653e-03 -3.535887e-03 0.0038245588 157.596319
28 2.211071e-03 -1.569349e-03 0.0027114003 125.365937
29 5.810030e-04 1.812466e-03 0.0019033121 17.773741

>

> period_dst\$ro

[1] 0.0037078813 0.0023564379 0.0027637869 0.0037725603
0.0016623073

[6] 0.0013454750 0.0014554103 0.0046411966 0.0034042215
0.0009255270

[11] 0.0033365439 0.0009987198 0.0025886635 0.0033647236
0.0003115571

[16] 0.0024957766 0.0025282054 0.0023683246 0.0012328117
0.0002662569

[21] 0.0011214332 0.0009949619 0.0003552106 0.0022915706
0.0011792845

[26] 0.0017311329 0.0038245588 0.0027114003 0.0019033121

>

```
> plot(period_dst$ro,type="l",main="Period.destag. GRAF.N.4, b")
```

Aspetto per confermare cambio pagina...

```
>
```

```
>
```

```
>
```

```
> #PROVIAMO INVECE A TOGLIERE IL TREND DALLA dst
```

```
>
```

```
> plot(dst,type="l", main="yt-destagionalizzata") # la yt o dst: yt  
destagionalizzata= ciclo+TREND +random (GRAF. N.4, a), anche ottenuti col  
programma Excel)
```

Aspetto per confermare cambio pagina...

```
>
```

```
> acf(dst, main="dst-CORR-yt-destag")
```

Aspetto per confermare cambio pagina...

```
>
```

```
> # da fare il periodogramma di dst ,(GRAF. N.4, b)
```

```
> # Se elimino il TREND da dst ottengo CLRD e posso controllare con CORR  
se
```

```
> # ciò che resta è da considerare residuo. yt-ESAs-TREND = CLRD
```

```
> # CLRD =yt-TREND- ESAs
```

```
> #Calcolo il trend di dst per toglierlo da yt-ESAs ed ottenere CLRD
```

```
> t=seq(1:60)
```

```
> fitdst=lm(dst~t)
```

```
> abline(lm(dst~t))
```

```
> summary(fitdst)
```

Call:

```
lm(formula = dst ~ t)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

-0.0244860 -0.0056415 -0.0008578 0.0074838 0.0230127

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.623e-04 2.460e-03 0.107 0.915

t 8.385e-06 7.013e-05 0.120 0.905

Residual standard error: 0.009407 on 58 degrees of freedom

Multiple R-squared: 0.0002464, Adjusted R-squared: -0.01699

F-statistic: 0.01429 on 1 and 58 DF, p-value: 0.9052

>

> resid(fitdst)

1 2 3 4 5

-0.0205850503 0.0114888567 -0.0018164030 0.0111442749

0.0003397444

6 7 8 9 10

0.0035515681 -0.0121568166 -0.0244860346 -0.0056090025

0.0080438628

11 12 13 14 15

0.0099242282 0.0022635518 -0.0066856661 -0.0115117591 0.0073829812

16 17 18 19 20

-0.0005563409 0.0156391286 -0.0052090477 0.0091425676

0.0098133496

21 22 23 24 25

-0.0014096184 -0.0176467530 -0.0087763877 0.0014629360

-0.0093862820

26 27 28 29 30

-0.0027123750 -0.0033176346 0.0036430432 -0.0070814873

-0.0029496636

31 32 33 34 35

```

0.0048419518 0.0230127338 0.0162897658 0.0035526311 0.0045229965
      36      37      38      39      40
0.0013623202 0.0091131022 0.0098870092 0.0029817496
0.0044624274
      41      42      43      44      45
-0.0118621031 0.0162497206 0.0007413359 -0.0046878820
-0.0058108500
      46      47      48      49      50
0.0088520153 -0.0032776193 -0.0029382957 0.0091124864
0.0077863934
      51      52      53      54      55
-0.0027188663 -0.0037581884 0.0054372811 -0.0073708952
-0.0011592799
      56      57      58      59      60
-0.0043884979 -0.0046114659 -0.0058486005 -0.0159782352
-0.0057389115
>
> p=predict(fitdst,data.frame(t=c(1,60)))
> CLRD=c()
> CLRD=(dst-p)
> CLRD
[1] -0.0205850503 0.0110025468 -0.0017996337 0.0106747343
0.0003732830
[6] 0.0030987968 -0.0121065087 -0.0249220365 -0.0055419253
0.0076246302
[11] 0.0100080747 0.0018610885 -0.0065850503 -0.0118974532
0.0075003663
[16] -0.0009252657 0.0157732830 -0.0055612032 0.0092934913
0.0094779635
[21] -0.0012419253 -0.0179653698 -0.0085919253 0.0011610885
-0.0091850503
[26] -0.0029974532 -0.0030996337 0.0033747343 -0.0068467170
-0.0032012032

```

```

[31] 0.0050934913 0.0227779635 0.0165580747 0.0033346302
0.0048080747
[36] 0.0011610885 0.0094149497 0.0097025468 0.0033003663
0.0042947343
[41] -0.0115267170 0.0160987968 0.0010934913 -0.0048220365
-0.0054419253
[46] 0.0087346302 -0.0028919253 -0.0030389115 0.0095149497
0.0077025468
[51] -0.0022996337 -0.0038252657 0.0058732830 -0.0074212032
-0.0007065087
[56] -0.0044220365 -0.0041419253 -0.0058653698 -0.0154919253
-0.0057389115
>
> CLRD=yt-ESAs-p
>
> n1=length(p)
>
> ts.plot(CLRD, main="GRAF. N.5-RESIDUI" )
Aspetto per confermare cambio pagina...
>
> acf(CLRD, main="GRAF. N.5-CORR_RESIDUI")
Aspetto per confermare cambio pagina...
>
> #periodogramma di CLRD
>
> yr=CLRD
>
> nr=length(yx)
>
> if (nr/2== nr%%2) mr=nr/2-1 else mr=(nr-1)/2
>
> period_clrd=PRDGRAM(yr, nr ,mr)

```

>

> period_clrd

	a	b	ro	phi
1	-3.204012e-03	-1.866197e-03	0.0037078813	239.781075
2	-1.284547e-03	1.975535e-03	0.0023564379	326.966987
3	-9.217332e-04	-2.605557e-03	0.0027637869	199.481576
4	4.871699e-04	3.740973e-03	0.0037725603	7.419616
5	-1.328386e-03	9.993281e-04	0.0016623073	306.953690
6	1.066929e-04	1.341238e-03	0.0013454750	4.548191
7	-8.956983e-04	1.147146e-03	0.0014554103	322.017071
8	-4.418815e-03	1.419430e-03	0.0046411966	287.808267
9	-1.869250e-03	-2.845106e-03	0.0034042215	213.305138
10	-9.179323e-04	-1.183237e-04	0.0009255270	262.654936
11	2.561849e-03	-2.137628e-03	0.0033365439	129.841895
12	-8.629213e-05	9.949849e-04	0.0009987198	355.043307
13	5.361577e-04	2.532531e-03	0.0025886635	11.953489
14	-2.923822e-03	-1.665122e-03	0.0033647236	240.338400
15	-1.766979e-04	-2.566042e-04	0.0003115571	214.551342
16	-1.569426e-03	1.940568e-03	0.0024957766	321.035940
17	1.970302e-03	-1.584214e-03	0.0025282054	128.800888
18	1.144974e-03	-2.073161e-03	0.0023683246	151.088829
19	-1.204297e-03	-2.636146e-04	0.0012328117	257.652985
20	1.105816e-04	-2.422075e-04	0.0002662569	155.460583
21	8.279972e-04	7.563287e-04	0.0011214332	47.590063
22	-9.924779e-04	-7.026236e-05	0.0009949619	265.950508
23	-3.479709e-04	7.134998e-05	0.0003552106	281.587646
24	1.545292e-03	-1.692149e-03	0.0022915706	137.597270
25	7.442295e-04	-9.147865e-04	0.0011792845	140.869734
26	1.468179e-03	9.172080e-04	0.0017311329	58.005960
27	1.457653e-03	-3.535887e-03	0.0038245588	157.596319

```

28 2.211071e-03 -1.569349e-03 0.0027114003 125.365937
29 5.810030e-04 1.812466e-03 0.0019033121 17.773741
>
> period_clrd$ro
[1] 0.0037078813 0.0023564379 0.0027637869 0.0037725603
0.0016623073
[6] 0.0013454750 0.0014554103 0.0046411966 0.0034042215
0.0009255270
[11] 0.0033365439 0.0009987198 0.0025886635 0.0033647236
0.0003115571
[16] 0.0024957766 0.0025282054 0.0023683246 0.0012328117
0.0002662569
[21] 0.0011214332 0.0009949619 0.0003552106 0.0022915706
0.0011792845
[26] 0.0017311329 0.0038245588 0.0027114003 0.0019033121
> plot(period_clrd$ro,type="l",main="Period.ciclo_random")
#da controllare ancora

>

```