

```

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)

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

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;b0=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")
#Per vedere i risultati trasferiti dalla console di R in pdf

```

```

#del precedente frammento di programma cliccare sotto:
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")

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")

#Mmesi0=c()

#a=mediamesi

#b=mean(as1)

#c=a-b

#Mmesi0=c() #12 valori medi meno la media serie originale; una specie di Effetto Stagionale

#Mmesi0=c(mediamesi - b)

#ts.plot(Mmesi0) # da controllare: Effetto Stagionale da confrontare con mediacol

#acf(Mmesi0, main="CORR_Mmesi0")

#Mmesi0 # 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 su as1 (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(n3%%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 Stagionali 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")
```

```
#ok
```

```
#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
CLRD
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")
# Appare chiaro che CLRD non sono random!!!!

```