

```

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 consolle 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")
```