

ΔΙΑΧΕΙΡΙΣΗ ΥΔΑΤΙΚΩΝ ΠΟΡΩΝ - ΕΡΓΑΣΤΗΡΙΟ

ΑΞΙΟΛΟΓΗΣΗ ΔΙΔΑΣΚΟΝΤΩΝ ΚΑΙ ΜΑΘΗΜΑΤΟΣ

<https://modip-server.kom.duth.gr/login.xhtml>

ΜΟΔΙΠ ΔΠΘ – Οδηγός αξιολόγησης μαθήματος από Φοιτήτριες - Φοιτητές



Περιβάλλον Πληροφοριακού Συστήματος ΜΟΔΙΠ

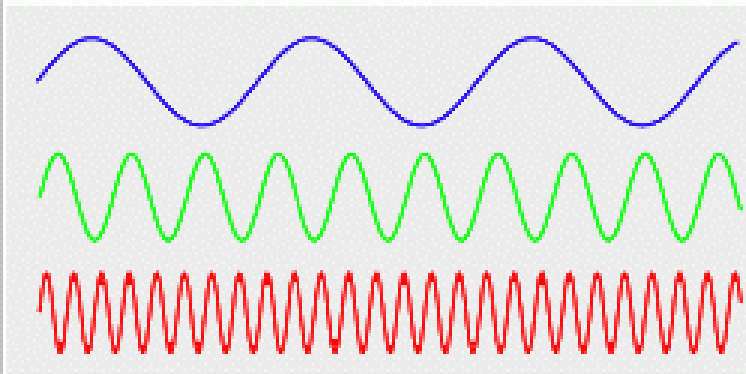
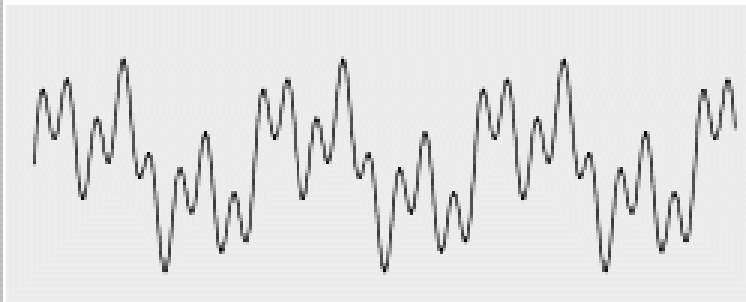
Για την συμμετοχή σας στην διαδικασία της ηλεκτρονικής αξιολόγησης μαθημάτων μέσω του Πληροφοριακού Συστήματος της ΜΟΔΙΠ (<http://modip-server.kom.duth.gr>), από την Αρχική οθόνη και χωρίς να συμπληρώσετε Όνομα χρήστη και Κωδικό ασφαλείας επιλέξτε από το μενού την επιλογή «Συμπλήρωση Ερωτηματολογίων» (Εικόνα 1).

The screenshot displays the MODIP system interface. At the top, there is a navigation bar with the following items: Αρχική, Συμπλήρωση Ερωτηματολογίων, and Δημοσιεύσεις. Below the navigation bar, there is a main content area with a header that reads "Μ.Ο.Δ.Ι.Π. ΔΗΜΟΚΡΕΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ" and "Πληροφοριακό Σύστημα Μονάδας Διασφάλισης Ποιότητας". The main content area is divided into two columns. The left column contains a "Διαπιστευτήρια" section with a sub-header "Πληκτρολογήστε τα στοιχεία σύνδεσης που σας έχουν δοθεί για να συνδεθείτε στο σύστημα." and a login form with fields for "Όνομα Χρήστη:" and "Κωδικός ασφαλείας:", and a "Είσοδος" button. The right column contains a "Νέα Ανακοινώσεις" section with a sub-header "Νέα Ανακοινώσεις της ΜΟΔΙΠ του Δημοκρίτειου Πανεπιστημίου Θράκης" and a "Συχνές ερωτήσεις" section with two links: "Οδηγός ενεργοποίησης αξιολόγησης μαθήματος από διδασκόμενες/ους" and "Οδηγός αξιολόγησης μαθήματος από φοιτήτριες - φοιτητές".

ΦΑΣΜΑΤΙΚΗ ΑΝΑΛΥΣΗ ΧΡΟΝΟΣΕΙΡΩΝ

- Φασματική Ανάλυση (Spectral Analysis) είναι η αποσύνθεση μίας χρονοσειράς σε ένα άθροισμα ημιτόνων και συνημιτόνων με διαφορετικές συχνότητες το κάθε ένα, η οποία μας επιτρέπει να καταλάβουμε ποιες συχνότητες, και άρα περιοδικότητες, εμφανίζουν σημαντική διακύμανση στην χρονοσειρά.

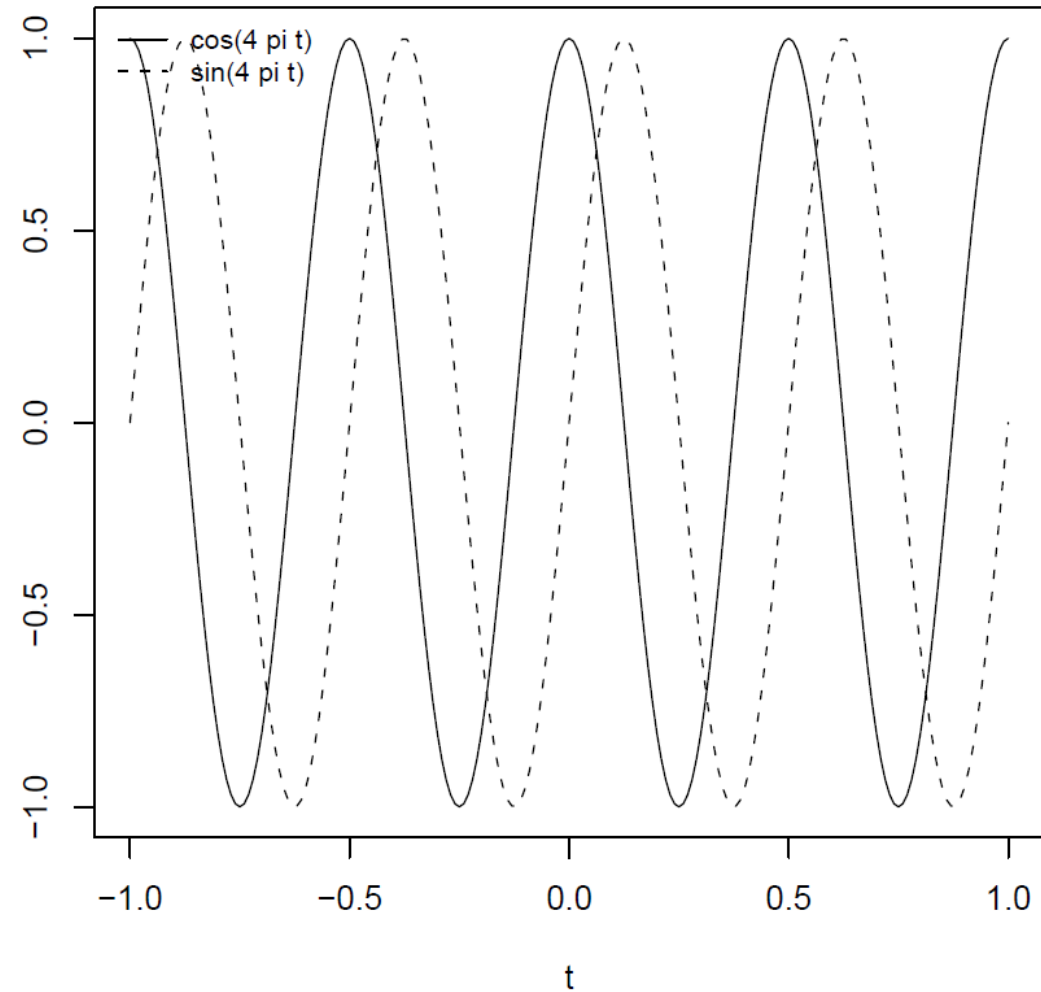
Fourier Analysis



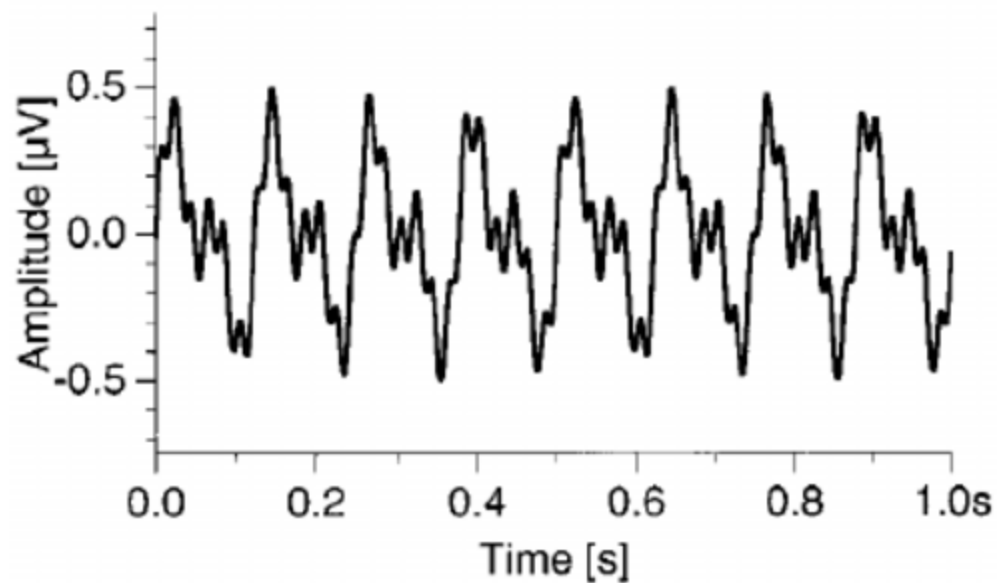
Fourier Analysis:
The complex wave (upper) can be decomposed into the sum of the three simple sine waves (lower).

- Η συχνότητα (f) των ημιτόνων και συνημιτόνων εκφράζεται σε κύκλους ανά μονάδα χρόνου. Στο παράδειγμα, η συχνότητα είναι 2 κύκλοι ανά μονάδα χρόνου. Η Περίοδος (T) του ημιτόνου ή συνημιτόνου είναι ($T = 1/f$), άρα στο παράδειγμα είναι $T = 1/2$.

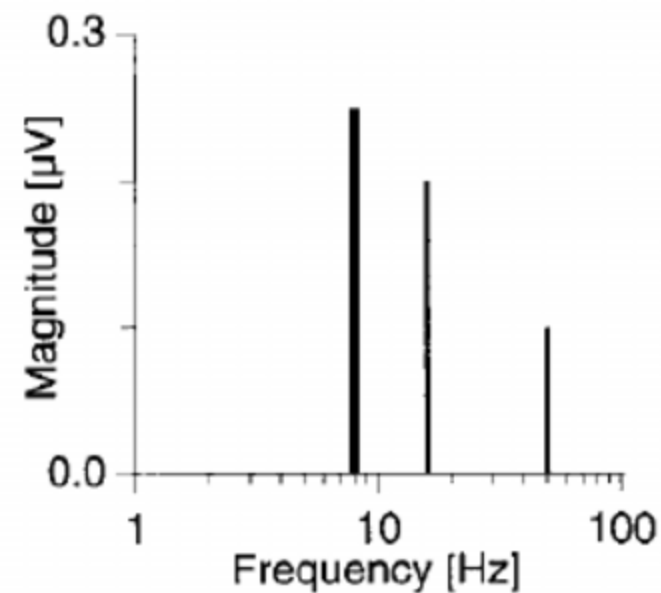
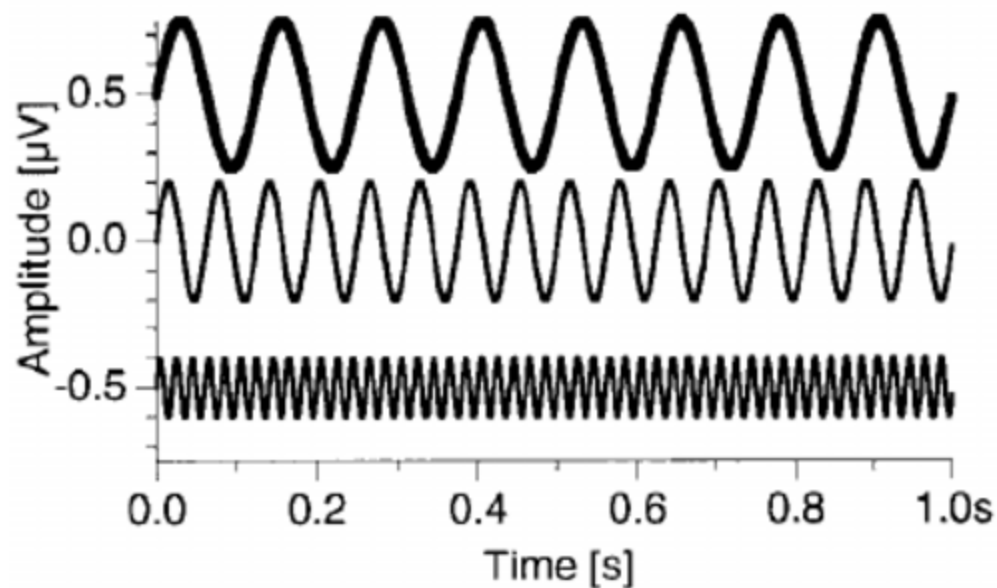
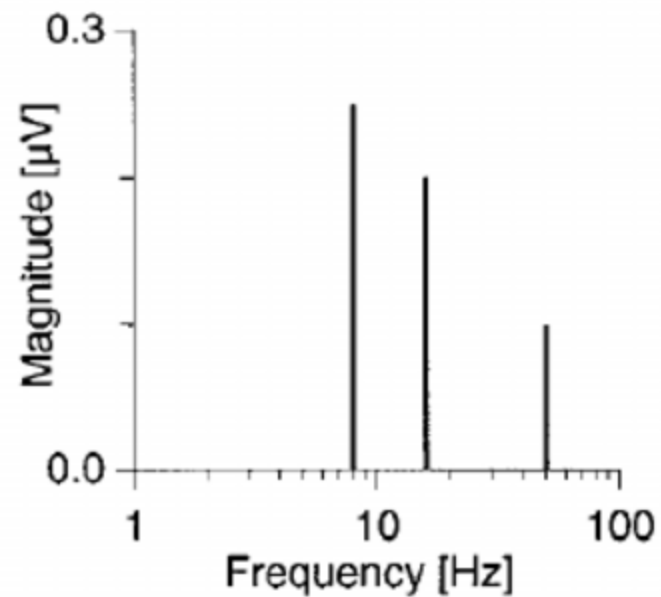
$$\cos\left(\frac{2\pi t}{T}\right), \sin\left(\frac{2\pi t}{T}\right)$$



Time series



Spectrum



Ένας τρόπος για καταλάβουμε την φασματική ανάλυση είναι να την δούμε σαν μία γραμμική παλινδρόμηση μεταξύ μίας εξαρτημένης μεταβλητής (την χρονοσειρά μετρήσεων) και μίας ανεξάρτητης μεταβλητής (ένα άθροισμα ημιτόνων και συνημιτόνων όλων των πιθανών, διακριτών συχνοτήτων).

Έστω ότι έχουμε μια χρονοσειρά Y_t με μήκος n (θεωρούμε ότι το n είναι περιττός). Μπορούμε να προσαρμόσουμε μία χρονοσειρά με παλινδρόμηση

$$y(t) = \overline{y(t)} + \sum_{p=1}^{N/2} \left[A_p \cos(\omega_p t) + B_p \sin(\omega_p t) \right] \quad \text{όπου } \omega_p = 2\pi f_p = 2\pi p/T$$

Οι συντελεστές συνημιτόνων, A_p , και οι συντελεστές ημιτόνων, B_p , μας εξηγούν τον βαθμό που ο συγκεκριμένος όρος συμβάλει στην βέλτιστη προσαρμογή στην χρονοσειρά.

Η μέθοδος αυτή καλείται **Ανάλυση Fourier** σε διακριτή χρονοσειρά.

Ο πρώτος όρος εκφράζει την μέση τιμή της χρονοσειράς.

Έστω μία χρονοσειρά $y(t)$ η οποία προέκυψε από δειγματοληψία ανά χρονικό βήμα Δt . Έστω ότι η χρονοσειρά έχει διάρκεια T , άρα αποτελείται από N χρονικά βήματα, ($N = T/\Delta t$) και από $N+1$ δεδομένα.

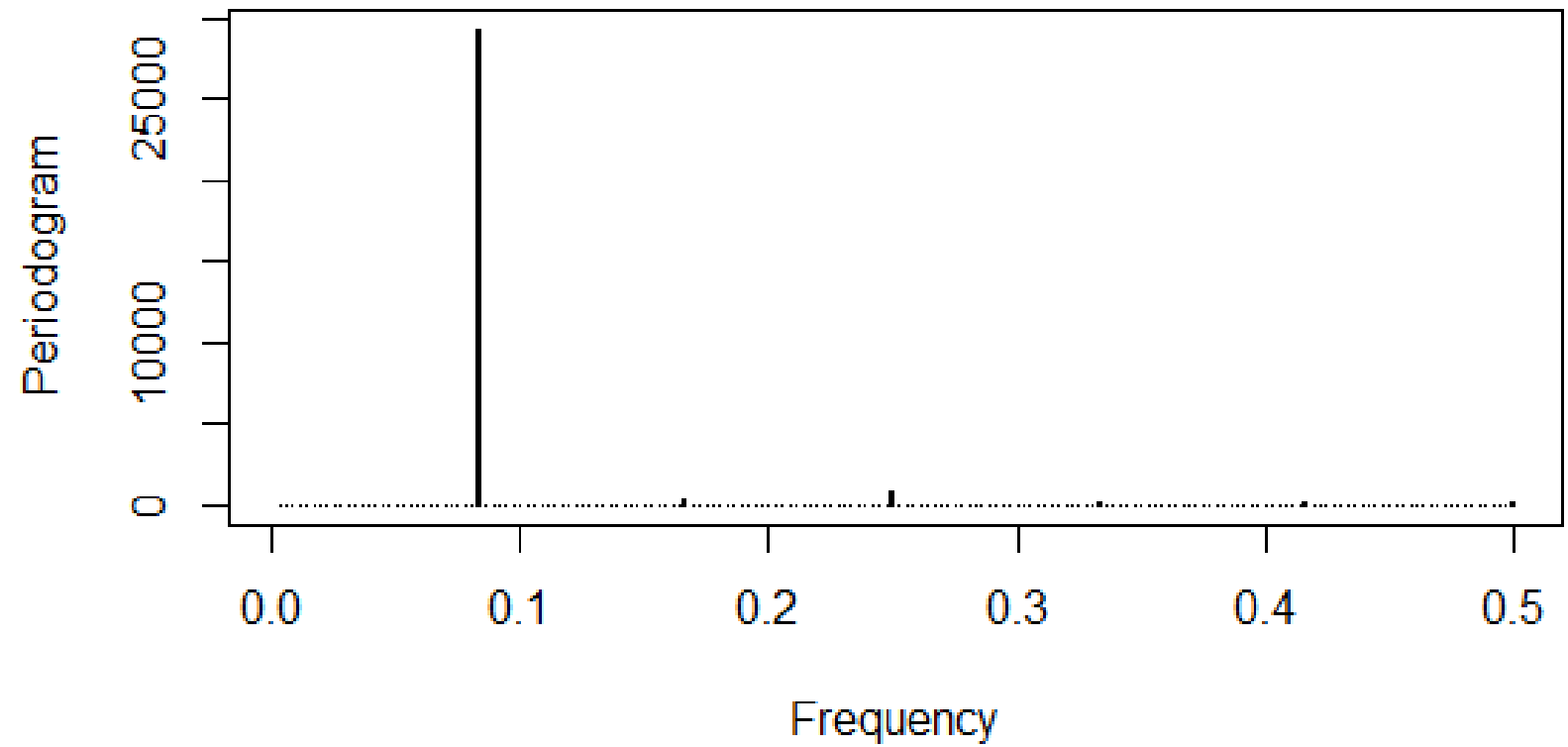
$$y(t_n) = y(n\Delta t) = y_n \quad (n = 0, 1, 2, \dots, N).$$

Με την ανάλυση Fourier θα αναπαράξουμε την χρονοσειρά ως ένα άθροισμα ημιτόνων και συνημιτόνων, δηλαδή κυμάτων με διαφορετικά εύρη και διαφορετικές συχνότητες. Τα κύματα αυτά καλούνται **αρμονικές (harmonics)** και αν αθροιστούν παράγουν την αρχική χρονοσειρά.

Έστω η συχνότητα $f=1/T, 2/T, 3/T, \dots$

κάθε αρμονική θα έχει την μορφή

$$y_k(n\Delta t) = C_k \cos[2\pi kn/N] \quad \text{όπου } C_k \text{ είναι το εύρος της κάθε αρμονικής.}$$



Η συνάρτηση `seasonality` επιστρέφει ένα πλαίσιο δεδομένων με μια στήλη που περιέχει τα ποσοστά της ετήσιας απορροής που αντιστοιχούν στην απορροή της μέσης απορροής των 6 ξηρότερων μηνών.

Επίσης επιστρέφει με το `monthly.means` μας δίνει την μέση απορροή κάθε μήνα του έτους.

Επίσης, επιστρέφει στο `avg.ann.month.range` την μέση διαφορά μεταξύ της μέγιστης και ελάχιστης απορροής.

Τέλος, το `max.min.time.dif` μας δίνει τον μέσο αριθμό μηνών μεταξύ της μέγιστης και της ελάχιστης απορροής.

```
setwd("C:/Users/user/OneDrive/Notes/Χανθι/ΔιαχείρισηΥδατικώνΠόρων")
```

```
install.packages(c("hydrostats","stats","utils","graphics",  
                  "dplyr","plyr","forecast","zoo"))
```

```
install.packages("hydroTSM")
```

```
library(hydrostats)
```

```
library(stats)
```

```
library(utils)
```

```
library(graphics)
```

```
library(dplyr)
```

```
library(plyr)
```

```
library(forecast)
```

```
library(zoo)
```

```
library(hydroTSM)
```

```
#Open the csv file into a dataframe
nestos = read.csv("NestosFlow.csv",
                 header = TRUE,
                 sep = ";")
#make a vector with Nestos flow data
nestos.flow = nestos$River.discharge..m3.s.

#Transform the vector into zoo object
dt <- seq(as.Date("1981-01-01"), as.Date("2010-12-31"), by = "days")
nestos.zoo = as.zoo(nestos.flow)
nestos.zoo = zoo(nestos.zoo,dt)

#Plot data with the hydroplot function
hydroplot(nestos.zoo,
          var.type="Flow",
          main="Randomized Data",
          pfreq = "dm")
```

```
#Get data summary
```

```
smry(nestos.zoo)
```

```
#Transform daily into monthly values
```

```
nestos.zoo.m <- daily2monthly(nestos.zoo, FUN=mean)
```

```
#Tranform monthly values from zoo to ts object
```

```
nestos.ts.m = ts(nestos.zoo.m, freq = 12, start=c(1981,1))
```

```
#Check the time-series
```

```
index(nestos.ts.m)
```

```
start(nestos.ts.m)
```

```
end(nestos.ts.m)
```

```
frequency(nestos.ts.m)
```

```
#decompose the time-series into T,S,R
```

```
nestos.stl = stl(nestos.ts.m, s.window="per")
```

```
plot(nestos.stl)
```

```
#Compute the relative strength of each component
```

```
Tt <- trendcycle(nestos.stl)
```

```
St <- seasonal(nestos.stl)
```

```
Rt <- remainder(nestos.stl)
```

```
#Calculate trend descriptive statistics
```

```
mean.Tt = mean(Tt)
```

```
median.Tt = median(Tt)
```

```
min.Tt = min(Tt)
```

```
max.Tt = max(Tt)
```

```
range.Tt = range(Tt)
```

```
range.Tt = range.Tt[2] - range.Tt[1]
```

```
summary(Tt)
```

```
#apply a linear model on daily time-series
```

```
t = seq(1,length(nestos.ts.m))
```

```
linear.model <- lm(Tt ~ t)
```

```
plot(t,Tt, type="l")
```

```
lines(linear.model$fitted.values, col="red")
```

```
#Trend Strength Calculation
```

```
Ft <- round(max(0,1 - (var(Rt)/var(Tt + Rt))),1)
```

```
#Seasonal Strength Calculation
```

```
Fs <- round(max(0,1 - (var(Rt)/var(St + Rt))),1)
```

```
#focus on the seasonal component
install.packages("TSA")
library(TSA)
per = periodogram(St,log='no',
  plot=TRUE,
  ylab="Periodogram",
  xlab="Frequency",lwd=2)
#Compute the main period of seasonality
1/per$freq[which.max(per$spec)]
#Take the main three frequencies of seasonality
spec1 = head(sort(per$spec, decreasing=TRUE), 3)
1/per$freq[which(per$spec==spec1[1])]
1/per$freq[which(per$spec==spec1[2])]
1/per$freq[which(per$spec==spec1[3])]

y.seasonal1 = cos(2*pi*t/12)+sin(2*pi*t/12)
y.seasonal2 = cos(2*pi*t/6) +sin(2*pi*t/6)
y.seasonal3 = cos(2*pi*t/4) +sin(2*pi*t/4)
y.seasonal = y.seasonal1+y.seasonal2+y.seasonal3
summary(fit <- lm(St~y.seasonal1+y.seasonal2+y.seasonal3))
plot(t,St, type="l")
lines(fitted(fit), col=2)
```

```
#transform time-series into a dataframe
```

```
dt1 = as.POSIXct(dt)
```

```
nestos.df = data.frame(dt1,nestos.zoo)
```

```
colnames(nestos.df) = c("Date","Q")
```

```
#This function takes a daily time series and returns
```

```
#the coefficient of variation of mean annual flow
```

```
#expressed as a percentage (sd/mean)*100
```

```
ann.cv(nestos.df)
```

```
#Calculate measure of central tendency
```

```
#and baseflow indices using the Lynne-Hollick filter
```

```
b0 = baseflows(nestos.df,a=0.975, ts="mean")
```

```
b1 = baseflows(nestos.df,a=0.975, ts="annual")
```

```
b2 = baseflows(nestos.df,a=0.975, ts="daily")
```

```
head(baseflows(nestos.df,a=0.975, ts="daily"))
```

```
#Make a daily plot of Q and baseflow
plot(nestos.df[,"Date"],nestos.df[,"Q"],
     type="l",
     xlab="Date",
     ylab="Discharge (ML/day)")
lines(b2$Date,b2$bf, col="red")
```

```
#Calculates the maximum flood length in a time series
flood.length.max(nestos.df, threshold = 150, ind.days = 5)
```

```
high.spell.lengths(nestos.df, threshold=150)
high.spells(nestos.df, quant=0.9)
```

```
#Calculate seasonality of discharge
seasonality(nestos.df, monthly.range=TRUE)
```

```
#Calculate the Cease-to-Flow statistics
CTF(nestos.df, threshold=6)
```

```
low.spell.lengths(nestos.df, threshold=6)
low.spells(nestos.df, quant=0.1)
```