#REGRESSIONE LINEARE SEMPLICE SU MODELLI DETERMINISTICI

##Un caso molto banale: regressione lineare su un modello deterministico lineare!

x=-2:2

y=3+x

plot(x,y)

fit = lm(y~x)

abline(fit, col='red')

summary(fit)

###R^2? p-values?

## pochi punti 'insoliti' potrebbero modificare molto la varianza spiegata R^2

x=-2:2

y=3+x

y[4]=-50

plot(x,y)

fit = lm(y~x)

abline(fit, col='red')

summary(fit)

###R^2? p-values?

##la 'varianza spiegata' misura solo correlazione lineare! è possibile che R^2 sia zero ma esiste una relazione di tipo non-lineare!

x= -5:5

y=x^2

plot(x,y)

fit = lm(y~x)

abline(fit, col='red')

summary(fit)

###R^2? p-values?

#REGRESSIONE LINEARE MULTIPLA

library('datasets')

###questa libreria contiene tanti dataset di esempio, non c'è bisogno di fare copy-paste da dati presi altrove: vediamo quali dataset ha caricato:

library(help = "datasets")

### ci sono diversi dataset interessanti

plot(sunspot.month)

plot(Titanic)

plot(presidents)

plot(quakes$lat, quakes$long)

### un altro esempio di regressione semplice

head(faithful)

plot(faithful)

fit = lm(faithful$eruptions ~ faithful$waiting)

abline(fit, col='red')

summary(fit)

## un esempio di regressione multipla: diametro (girth), altezza, volume di alberi di ciliegio

head(trees)

plot(trees)

fit = lm(formula = trees$Volume ~ trees$Girth + trees$Height)

summary(fit)

###ma questo non ha molto senso, no? piuttosto cerchiamo volume =c \* altezza^a \* diametro^b -> a=1, b=2

log\_trees = log(trees)

fit = lm(formula = log\_trees$Volume ~ log\_trees$Girth + log\_trees$Height)

summary(fit)

## un esempio di regressione multipla: Germany DAX (Ibis), Switzerland SMI, France CAC, and UK FTSE

head(EuStockMarkets)

indici = as.data.frame(EuStockMarkets)

plot(indici$FTSE, type='l')

plot(indici$DAX, type='l')

plot(log(indici$DAX), type='l')

log\_indici=log(indici)

### ancora regressione lineare semplice: fit lineare e fit 'esponenziale'

T=length(indici$FTSE)

tempo = 1:T

plot(tempo, indici$FTSE, type='l')

fit = lm(indici$FTSE ~ tempo)

abline(fit)

summary(fit)

plot(tempo, log\_indici$FTSE, type='l')

log\_fit = lm(log\_indici$FTSE ~ tempo)

abline(log\_fit)

summary(log\_fit)

### proviamo a vedere se r^2 cresce al crescere della numerosità del campione

summary( lm(log\_indici$FTSE[1:10]~ tempo[1:10]) )

summary( lm(log\_indici$FTSE[1:100]~ tempo[1:100]) )

summary( lm(log\_indici$FTSE[1:1000]~ tempo[1:1000]) )

#passiamo alla regressione multipla: ci aspettiamo grande correlazioni tra i vari indici europei

fit3=lm(log\_indici$FTSE ~ log\_indici$CAC + log\_indici$SMI+log\_indici$DAX)

summary(fit3)

fit2=lm(log\_indici$FTSE ~ log\_ indici$CAC + log\_indici$SMI)

summary(fit2)

fit1=lm(log\_indici$FTSE ~ log\_ indici$CAC)

summary(fit1)

#REGRESSIONE LINEARE MULTIPLA PER 'FIT' POLINOMIALE

##Abbiamo visto esempi in cui il modello diventava lineare dopo una trasformazione logaritmica (si dice passando a scala logaritmica). La regressione lineare multipla si può usare però anche per 'fittare' con funzioni non lineari, ad esempio polinomi. Facciamo un esempio deterministico:

x= -5:5

y=2+ 5\*x + 3\*x^2

plot(x,y)

fit = lm(y~x)

abline(fit, col='red')

summary(fit)

## chiaramente l'approssimazione lineare non è affatto buona. Possiamo però cercare un modello lineare della forma y = a + b\*x + c\*z, dove z = x^2.

z = x^2

fit = lm(y~x+z)

summary(fit)

curve(2+5\*x+3\*x^2, from=-5, to=5, add=TRUE, col='blue')

### se volessimo ottenere una approssimazione con polinomi di grado più alto cosa accadrebbe?

x2=x^2

x3=x^3

x4=x^4

fit = lm(y~x+x2+x3)

### ATTENZIONE: se cresce il grado del polinomio, ossia il numero di parametri, abbiamo solo un'illusione di miglior approssimazione:

y =2+ 5\*x + 3\*x^2 + 5\*rnorm(length(x))

plot(x,y)

z = x^2

fit = lm(y~x+z)

summary(fit)

a= fit

curve( a+b\*x+c\*x^2, from=-5, to=5, add=TRUE, col='blue')

x2=x^2

x3=x^3

x4=x^4

fit = lm(y~x+x2+x3+x4)

summary(fit)