Tp3 : programmation par Matlab des lois statistique et problème interpolation

Lire des données sous forme d'un tableau	`x=[43 43 43 47 48 48 48 48 49 49 49 50 50 51 51 ... 52 53 53 53 54 54 56 56 56 57 59 59 59 62 62 63 ... 63 65 65 67 67 68 70 70 70 72 72 73 77 77 81 83 ... 86 92 93]`	
Paramètres de position		
Nombre des éléments	N=length(x)	
La moyenne algébrique : $\bar{x} = \frac{1}{N}.\sum_{I=1}^{N} x_I$	`X_moy=sum(x)/N`	`X_moy=mean(x)`
Moyenne géométrique : $G = \left(\prod_{I=1}^{N} x_I\right)^{\frac{1}{N}}$	`Moy_g=prod(x)^(1/N)`	`Moy_g=geomean(x)`
Moyenne harmonique : $H = \frac{N}{\sum_{I=1}^{N} \frac{1}{x_I}}$	`Moy_H=N/sum(1./x)`	`Moy_H=harmmean(x)`
Quantiles x_P : le premier quartile : $Q1 = x_{0.25}$ deuxième quartile : $Q2 = x_{0.5}$ (médiane) le troisième quartile : $Q3 = x_{0.75}$	`Q1=quantile(x,0.25)` `Q2=quantile(x,0.5)` `Q3=quantile(x,0.75)`	

Paramètres de dispersion

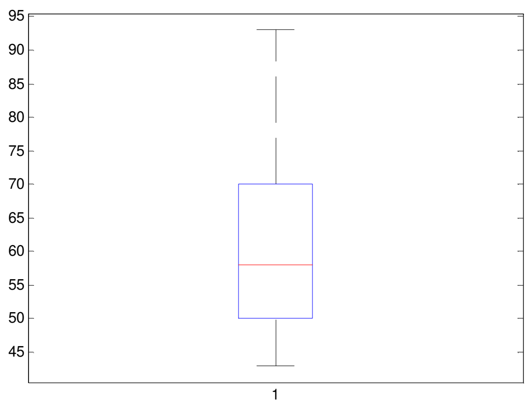
L'étendue : E=max(x)-min(x)	**E=max(x)-min(x)**	
La distance interquartile : $IQ = Q3 - Q1$	`IQ=quantile(x,0.75)-quantile(x,0.25)`	
La variance : $S_x^2 = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I - \bar{x}\right)^2 = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I\right)^2 - \bar{x}^2$	`var=sum((x-mean(x)).^2)/N`	`Var(x)`
L'écart-type : $S_x = \sqrt{\frac{1}{N}.\sum_{I=1}^{N}\left(x_I - \bar{x}\right)^2}$	`ecart_type1=sqrt(var)`	`std(x)`
L'écart-type corrigée : $(S_x)_C = S_x.\sqrt{\frac{N}{N-1}}$	`sqrt(var1)*sqrt(N/(N-1))`	`std(x)`
L'écart moyen absolu $emoy = \frac{1}{N}.\sum_{I=1}^{N}\left(\left\|x_I - \bar{x}\right\|\right)$	`emoy=sum(abs(x-mean(x)))/N`	
L'écart médian absolu $E_{med} = \frac{1}{N}.\sum_{I=1}^{N}\left(\left\|x_I - x_{1/2}\right\|\right)$	`emed=sum(abs(x-median(x)))/N`	

Paramètres de forme

Moments : Moment à l'origine : $m_r = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I^r\right)$; **Moment à centré:** $m_r = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I - \bar{x}\right)^r$	
Moment à centré d'ordre 3 : $m_3 = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I - \bar{x}\right)^3$ **Coefficient d'asymétrie de Fisher** $g_1 = \frac{m_3}{S_x^3}$	`m3=sum((x-mean(x)).^3)/N` `g1=m3/std(x).^3`

Coefficient d'asymétrie de Yule : $$A_Y = \frac{x_{0.75} + x_{0.25} - 2.x_{0.5}}{x_{0.75} - x_{0.25}}$$	Q1=**quantile**(x,0.25) Q2=**quantile**(x,0.5) Q3=**quantile**(x,0.75) ay=(Q3+Q1-2*Q2)/(Q3-Q1)
Coefficient d'asymétrie de Pearson : $$A_P = \frac{\bar{x} - x_M}{S_x} \quad ; \quad x_M \text{ : Le mode}$$	ap=(**mean**(x)-**median**(x))/**std**(x)
Paramètre d'aplatissement : coefficient d'aplatissement de Pearson $$\beta_2 = \frac{m_4}{S_x^4}$$ coefficient d'aplatissement de Fisher $$g_2 = \beta_2 - 3$$	m4=**sum**((x-**mean**(x)).^4)/N bita2=m4/(**std**(x).^4) g2=bita2-3

Présentation graphique

La boıte à moustaches est un diagramme qui permet de représenter la distribution d'une variable x. Max Q3 Médiane Q1 Min	boxplot(x) 

Statistique descriptive bi variée

Lire des données sous forme d'un tableau	x=[155,162,157,170,164,162,169,170,178,173, … 180, 175, 173,175,179,175,180,185,189,187] y= [60 ,61 ,64 ,67 ,68 ,69 ,70 ,70 ,72 ,73 , … 75 ,76 ,78 ,80 ,85 ,90 ,96 ,96 ,98 ,101]

Analyse des variables Deux variables quantitatives

paramètres marginaux *la moyenne, variance et écarte type* $$\bar{x} = \frac{1}{N}.\sum_{I=1}^{N} x_I \quad ; \quad S_x^2 = \frac{1}{N}.\sum_{I=1}^{N}\left(x_I^{\,2}\right) - \bar{x}^2$$ $$\bar{y} = \frac{1}{N}.\sum_{I=1}^{N} y_I \quad ; \quad S_y^2 = \frac{1}{N}.\sum_{I=1}^{N}\left(y_I^{\,2}\right) - \bar{y}^2$$	xm=**sum**(x)/n ym=**sum**(y)/n var_x=**sum**(x.^2)/n-xm^2 var_y=**sum**(y.^2)/n-ym^2	xm=**mean**(x) ym=**mean**(y) var_x=**var**(x) var_y=**var**(y)
Covariance : $S_{xy} = \frac{1}{N}.\sum_{I=1}^{N}(x_I.y_I) - \bar{x}.\bar{y}$	cov=**sum**(x.*y)/n-xm*ym	
Corrélation : $r_{xy} = \frac{S_{xy}}{S_x.S_y}$	Corr=**cov**/(**std**(x)***std**(y))	
coefficient de détermination : $r_{xy}^2 = \frac{S_{xy}^2}{S_x^2.S_y^2}$	Coef_det=corr.^2	

Interpolation un seul variable

En analyse numérique, **l'interpolation** est une opération mathématique permettant de construire une fonction à partir des données expérimentales $\left(x(k), f(k)\right)$.

$x(k)$	$x(1)$	$x(2)$	$x(3)$		$x(N)$
$f(k)$	$f(1)$	$f(2)$	$f(3)$		$f(N)$

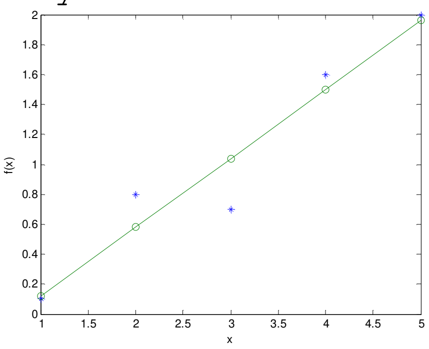
Dans l'objectif de déterminer les coefficients ***c*** de la fonction ***f(x)*** on utilisant la command :

$$c = \textbf{\textit{nlinfit}}(x, f, myfun, c0)$$

Tel que : c0 c'est les valeurs estime ; myfun : c'est la fonction proposé

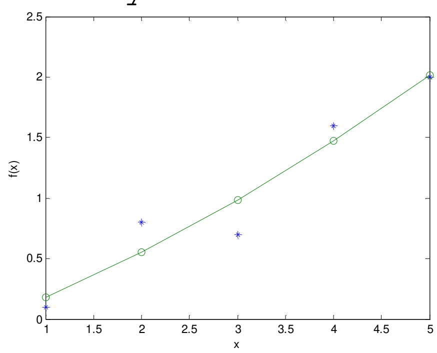
```
clc ; clear all
np=5 ;  %  nombre de point
x(1,1:np)=[1   2   3   4   5] ; f(1:np)  =[0.1 0.5 0.9 1.6 2 ] ;
%====== nc : nombre des coefficients dans la fonction f(x)
nc=2 ; myfun = @(c,x) c(1) + c(2)*x ;
%nc=3 ; myfun = @(c,x) c(1) + c(2)*x + c(3)*x.^2  ;
%nc=4 ; myfun = @(c,x) c(1) + c(2)*x + c(3)*x.^2  + c(4)*x.^3 ;
%nc=5 ; myfun = @(c,x) c(1) + c(2)*x + c(3)*x.^2  + c(4)*x.^3 + c(5)*x.^4;
%nc=3 ; myfun = @(c,x) c(1) + c(2)  * exp(c(3)*x) ;
%nc=3 ; myfun = @(c,x) c(1) + c(2)  * x.^c(3) ;
%nc=3 ; myfun = @(c,x) c(1) + c(2) *log( c(3).*x) ;
%======= calcule des coefficient  =========
c0(1:nc)= 1 ;    c = nlinfit(x,f,myfun,c0)
fnew = myfun(c,x) ;
%======= calcule   R^2  =========
r2=1-sum((f-fnew).^2)/sum((f-mean(fnew)).^2)
plot(x,f,'*',x, fnew,'-o')  ;   xlabel('x') ; ylabel('f(x)')
```

Polynôme ordre 1



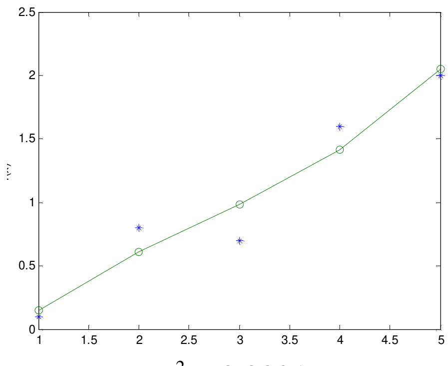
$R^2 = 0.9232$

Polynôme ordre 2



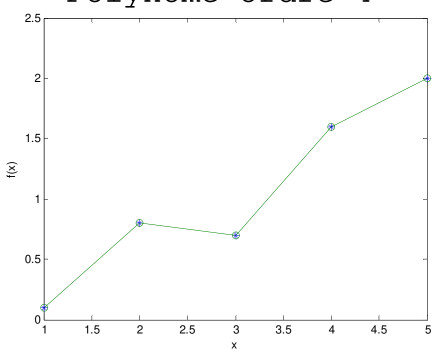
$R^2 = 0.9282$

Polynôme ordre 3



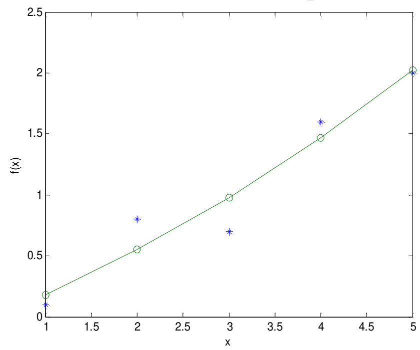
$R^2 = 0.9321$

Polynôme ordre 4



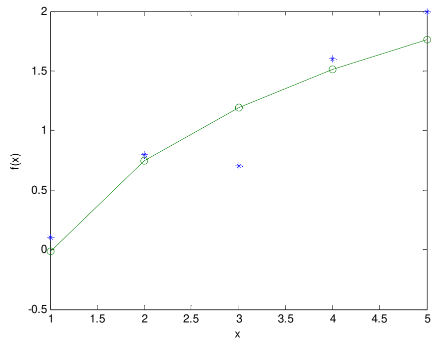
$R^2 = 1$

F(x)=c(1)+c(2)*exp(c(3)*x)



$R^2 = 0.9285$

F(x)=c(1)+c(2)*Ln(c(3)*x)



$R^2 = 0.8571$

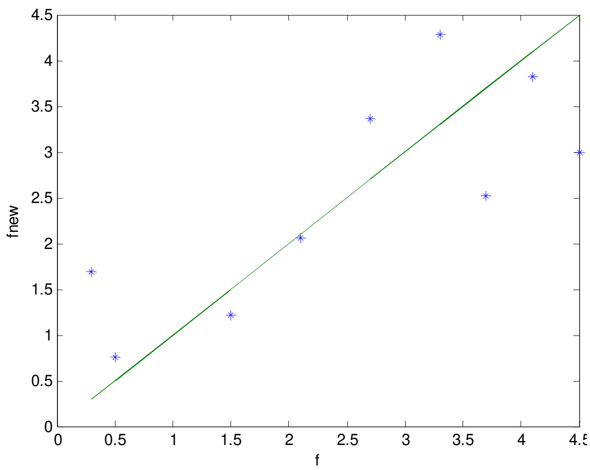
Remarque : on justifie la meilleure interpolation si on trouve $R^2 \to 1$

Interpolation à deux variables

On peut utiliser le même principe pour construire une fonction à deux variables ***f(x1, x2).*** On a choisi des exemples, le programme doit être modifié comme suit :

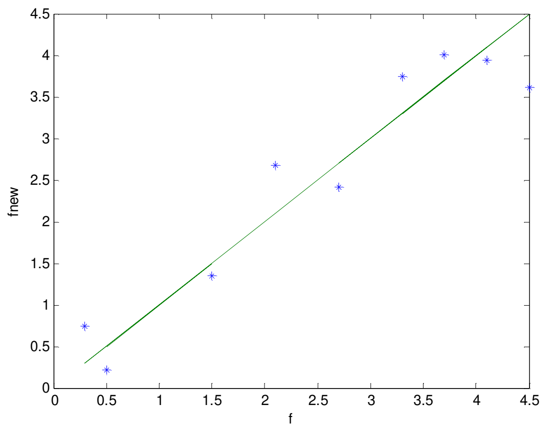
```
clc ; clear all
np=9 ;  %  nombre de point

x(1,1:np)=[1    1    1    2     2    2    3    3     3]
x(2,1:np)=[1    2    3    1     2    3    1    2     3]
f(1:np)  =[0.5  1.5  0.3  2.1   3.7  4.5  2.7  4.1   3.3 ]'
%====== nc : nombre des coefficients dans la fonction f
 nc=3 ; myfun= @(c,x) c(1) + c(2)*x(1,:)+ c(3)*x(2,:) ;
% nc=6 ; myfun= @(c,x) c(1) + c(2)*x(1,:)+ c(3)*x(2,:)+ ...
% c(4)*x(1,:).*x(1,:)+c(5)*x(2,:).*x(2,:)+ c(6)*x(1,:).*x(2,:) ;
%======= calcule des coefficient  =========
c0(1:nc)= 1 ;   c = nlinfit(x,f,myfun,c0) ;
fnew = myfun(c,x)
%======= calcule R^2  =========
r2=1.-sum((f-fnew).^2)/sum((f-mean(fnew)).^2)
 plot(f,fnew,'*',f,f)  ; xlabel('f') ; ylabel('fnew')
```

F(x)=c(1)+c(2)*x1+c(3)*x2



$R^2 = 0.6129$

F(x)=c(1)+c(2)*x1+c(3)*x2+
c(4)*x1*x1+c(5)*x2*x2+c(6)*x1*x2



$R^2 = 0.902$

Remarque : lorsque il y a une difficulté de compare les courbes en plusieurs variables, Dans ce cas on a choisi de tracer la courbe (f, fnew)