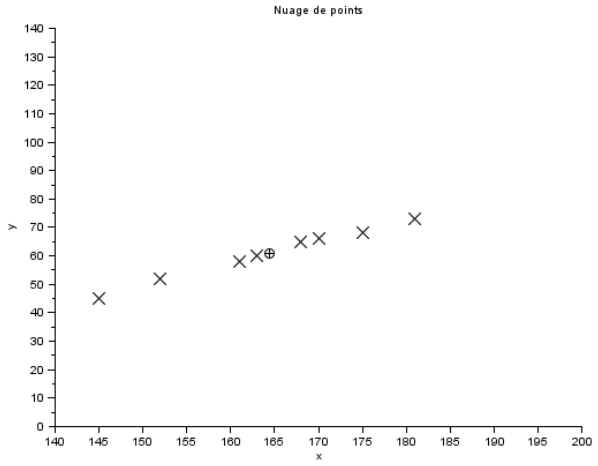
Correction du TP6

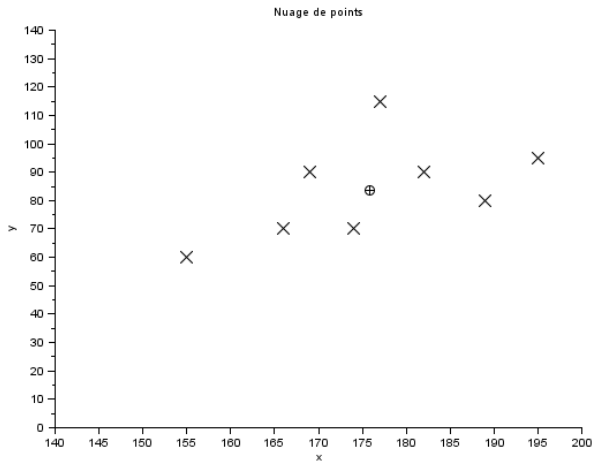
Exercice 1.

1. A priori le poids dépend de la taille donc la variable explicative est X et la variable à expliquer est Y.
2.
```
-->// série statistique des tailles:

-->x=[161 170 152 181 163 145 168 175]
 x  =

    161.    170.    152.    181.    163.    145.    168.    175.

-->// série statistique des poids :

-->y=[58 66 52 73 60 45 65 68]
 y  =

    58.    66.    52.    73.    60.    45.    65.    68.

-->// Tracé du nuage de points :

-->plot2d(x,y,style=-2)

-->// Point moyen:

 -->plot2d(mean(x), mean(y), style=-3)

-->// titre et étiquetage des axes:

-->xtitle("Nuage de points","x","y")

--> // changement des bornes: le numéro 1 indique xmin, 2 indique xmax, 3 indique ymin, 4 indique ymax :

-->a=get("current_axes") ; a.data_bounds([1 2 3 4])=[140 200 0 140];
```



On remarque qu'il y a une forte corrélation linéaire entre la taille X et le poids Y, c'est-à-dire qu'il y a une relation du type $Y \simeq aX + b$ ou plus exactement $Y = aX + b + \varepsilon$ où ε est faible.
3.
```
-->// série statistique des tailles:

-->x=[169 195 177 182 166 155 189 174]
 x  =
```

```
    169.    195.    177.    182.    166.    155.    189.    174.

-->// série statistique des poids :

-->y=[90 95 115 90 70 60 80 70 ]
 y  =

    90.    95.    115.    90.    70.    60.    80.    70.

-->// Tracé du nuage de points avec les mêmes bornes que précédemment:

-->scf(1), plot2d(x,y,style=-2)

-->// Point moyen:

 -->plot2d(mean(x), mean(y), style=-3)

-->// titre et étiquetage des axes:

-->xtitle("Nuage de points","x","y")

--> // changement des bornes: le numéro 1 indique xmin, 2 indique xmax, 3 indique ymin, 4 indique ymax :
```



Il est difficile de voir apparaître une fonction qui relie X et Y.
On remarque qu'il y a tout de même une variation selon une certaine direction oblique, on a donc $Y = aX + B + \varepsilon$ mais l'erreur d'ajustement ε est forte.
Il y a donc une faible corrélation entre X et Y.

4. Aux U.S.A., il y a un pourcentage important de personnes au comportement alimentaire déréglé.....Ceux qui ne mangent pas à leur faim (trop ou insuffisamment) ont un poids peu corrélé à leur taille.

Exercice 2.

1.
```
-->// Série statistique x:

-->x=[1:20]+grand(1,20,'unf',-1/2,1/2)          // remarque: on pouvait aussi utiliser une boucle for

 x  =


        column 1 to 8

   1.0094561    2.1343169    3.120798    3.507603    5.2335674    5.6084528    6.7299962    8.0578314

        column  9 to 16
```

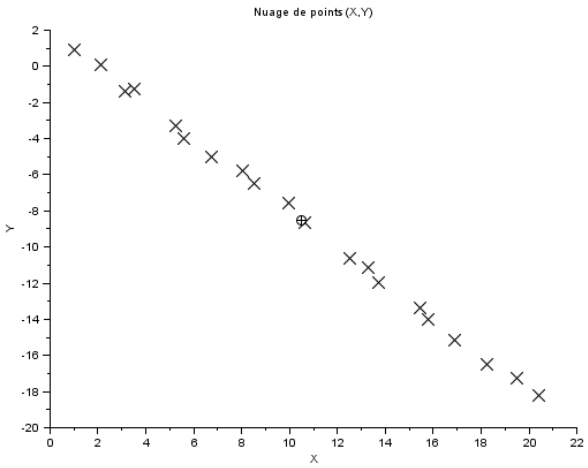
```
    8.5218719    9.9634339    10.638989    12.492742    13.269507    13.703227    15.469765    15.775001

         column 17 to 20

    16.886794    18.222396    19.493428    20.378619
```

2. `-->// série statistique z:`

```
-->z=grand(1,20,'nor',0,0.25)
 z =


         column 1 to 8

  - 0.1015735    0.2267106  - 0.2552437    0.2452957  - 0.0630375  - 0.3708113  - 0.2611031    0.2591092

         column  9 to 16

    0.0479709    0.3885598  - 0.0201127  - 0.1095332    0.1466023  - 0.2335899    0.1027872  - 0.239571

         column 17 to 20

  - 0.2630621  - 0.2554881    0.2114288    0.1883343
```

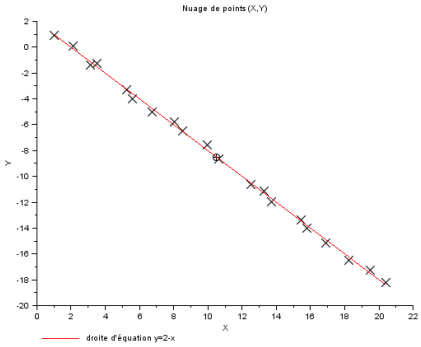
3. `--> // Nuage de points de la série statistique associée au couple (X,Y):`

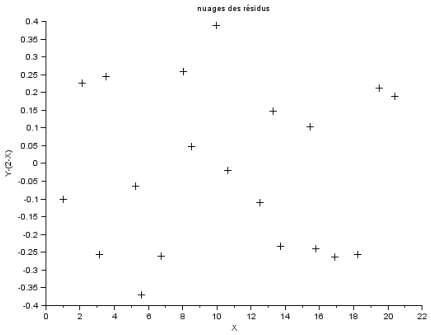
```
-->y=2-x+z; plot2d(x,y,style=-2)

-->// Point moyen:

-->plot2d(mean(x), mean(y), style=-3)

-->xtitle("Nuage de points  (x,y)", "x", "y")
```



On remarque une corrélation affine entre X et Y donc $Y \simeq aX + b$.
En lisant l'ordonnée à l'origine on prend $b = 2$, en lisant la pente de la courbe on prend $a = -1$.
On a donc $Y \simeq 2 - X$

```
 --> //Tracé de la courbe de f(x)=2-x en rouge, avec une légende:

-->plot2d(x,2-x,style=5, leg='droite d''équation y=2-x')

--> // Tracé des résidus (erreurs d'ajustement):

--> scf(1),plot2d(x, y-(2-x),style=-1)
```





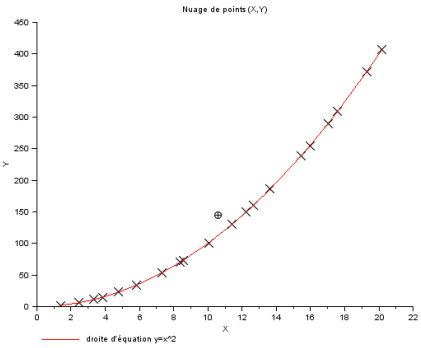
On remarque sur le premier graphique que le nuage de point est "pratiquement" sur la droite d'équation $y = 2 - x$.
On a donc $Y = 2 - X + \varepsilon$ avec ε petit :
Sur le second graphique, on trace ε en fonction de X. On remarque alors que l'erreur résiduelle vaut 0 en moyenne et que le nuage de points (X, ε) ne se répartit pas autour d'une direction précise.La régression choisie est donc une bonne régression.

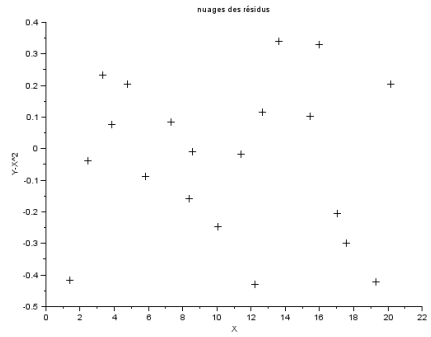
```
--> // Calcul de la covariance:

-->cov=mean(x.*y)-mean(x)*mean(y)
 cov  =

  - 34.627702
```

La covariance est négative ce qui est cohérent car globalement Y décroît lorsque X croît (Y vaut à peu près une fonction décroissante de X).

4. • On remplace $2 - X$ par X^2 dans le programme précédent :

```
// Série statistique x:
x=[1:20]+grand(1,20,'unf',-1/2,1/2)
// série statistique z:
z=grand(1,20,'nor',0,0.25)
// Nuage de points (x,y):
y=x.^2+z; plot2d(X,Y,style=-2)
// Point moyen:
plot2d(mean(x), mean(y), style=-3)
xtitle("Nuage de points (x,y)", "x", "y")
//Tracé de la courbe de f(x)=x^2 en rouge, avec une légende:
plot2d(x,x.^2,style=5, leg='droite d''équation y=x^2')
// Tracé des résidus (erreurs d'ajustement):
scf(1),plot2d(x, y-x.^2,style=-1)
xtitle("nuages des résidus","x","y-x^2")
cov=mean(x.*y)-mean(x)*mean(y)
disp(cov)
```



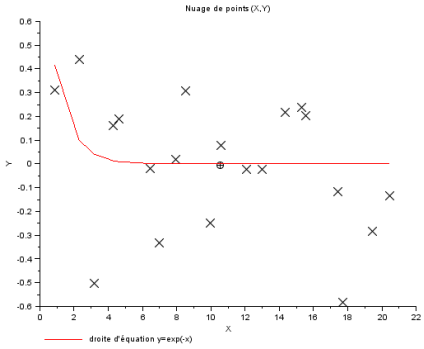


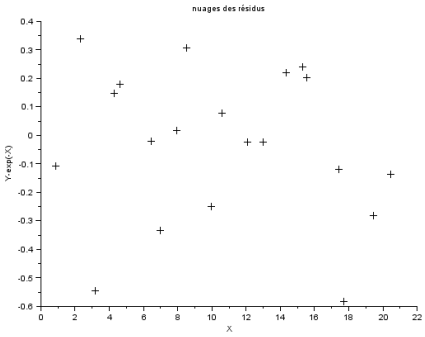
Les résultats sont les même c'est-à-dire qu'on a une bonne corrélation. En revanche, la covariance est ici positive ce qui est logique car $Y \simeq X^2$ croît quand X croît.

• On remplace $2 - X$ par $exp(-X)$ dans le programme précédent :

```
// Série statistique x:
x=[1:20]+grand(1,20,'unf',-1/2,1/2)
// série statistique z:
```

```
z=grand(1,20,'nor',0,0.25)
// Nuage de points (x,y):
y=exp(-x)+z; plot2d(x,y,style=-2)
// Point moyen:
plot2d(mean(x), mean(y), style=-3)
xtitle("Nuage de points (x,y)", "x", "y")
//Tracé de la courbe de f(x)=exp(-x) en rouge, avec une légende:
plot2d(x,exp(-x),style=5, leg='droite d''équation y=exp(-x)')
// Tracé des résidus (erreurs d'ajustement):
scf(1),plot2d(x, y-exp(-x),style=-1)
xtitle("nuages des résidus","x","y-exp(-x)")
cov=mean(x.*y)-mean(x)*mean(y)
disp(cov)
```

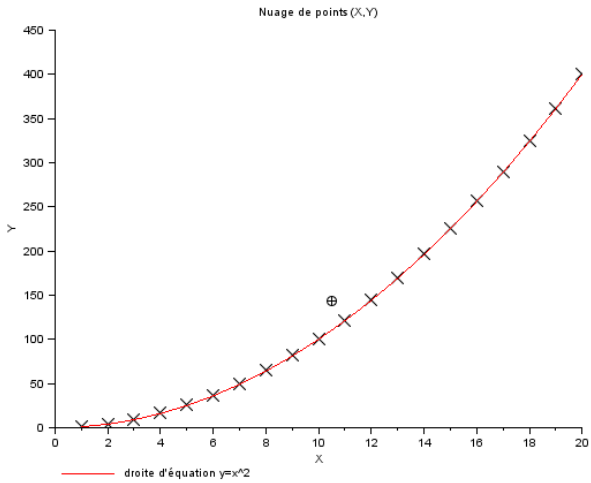




On remarque d'après le premier graphe que l'erreur résiduelle est trop forte. La corrélation est faible ici, il n'est pas judicieux de modéliser le lien entre x et y par l'équation $y = \exp(-x)$. La covariance est d'ailleurs négative mais très faible car la relation de décroissance entre X et Y est peu claire.

Exercice 3.

1.
```
// Série statistique x:
x=[1:20]
// série statistique y:
y=x.^2+rand(1,20)
// Nuage de points (x,y):
plot2d(x,y,style=-2)
// Point moyen:
plot2d(mean(x), mean(y), style=-3)
xtitle("Nuage de points (x,y)", "x", "y")
//Tracé de la courbe de f(x)=x^2 en rouge, avec une légende:
plot2d(x,x.^2,style=5, leg='droite d''équation y=x^2')
```

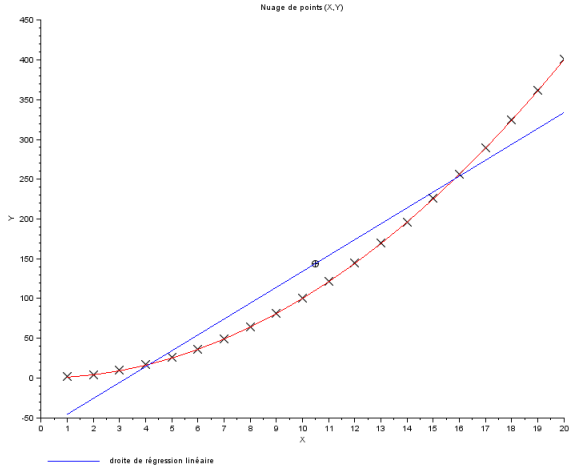


On remarque que l'équation $y = x^2$ est une bonne corrélation de Y en fonction de X.

On sait donc d'avance, au vu du comportement de la courbe, qu'on ne peut pas décrire correctement la relation entre X et Y par une équation affine.

2.
```
// Calculs des covariance et variances empiriques:
c=mean(x.*y)-mean(x)*mean(y)
v1=mean(x.^2)-mean(x)^2
v2=mean(y.^2)-mean(y)^2
```

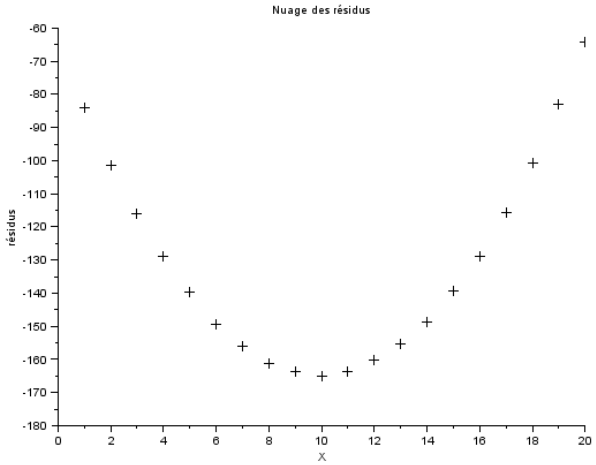
3.
```
// Calcul des coefficients de la droite de régression:
a=c/v1
disp(a,"a=")
b=mean(y)-a*mean(x)
disp(b,"b=")

// Tracé de la droite de régression linéaire:
plot2d(x,a*x+b,style=2, leg='droite de régression linéaire')
```




```
// Calcul du coefficient de corrélation linéaire:
rho=c/(sqrt(v1)*sqrt(v2))
disp(rho,"rho=")
```

La régression linéaire n'est pas une bonne approximation du nuage de points, pourtant le coefficient de corrélation linéaire est très proche de 1.
On remarque tout de même que le point moyen est sur cette droite.

4.
```
scf(1), plot2d(x, y-a*x-b,style=-1)
xtitle("Nuage des résidus","x","résidus")
```

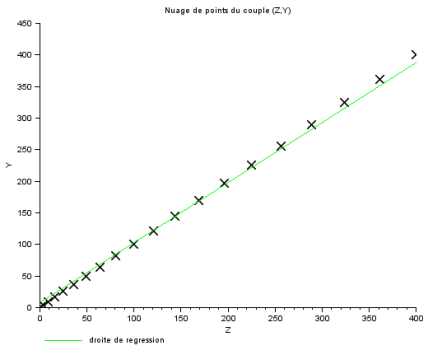


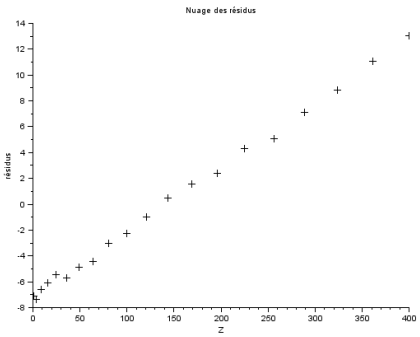
Le nuage des résidus a clairement une direction particulière. L'approximation affine est une très mauvaise corrélation.

5.
```
// Série statistique x:
x=[1:20]
```

```
// série statistique y:
y=x.^2+rand(1,20)

z=x.^2
// Nuage de points (z,y):
plot2d(z,y,style=-2)
xtitle("Nuage de points (z,y)", "z", "y")

// Calculs des covariance et variances empiriques:
c=mean(z.*y)-mean(z)*mean(y)
v1=mean(z.^2)-mean(z)^2
v2=mean(y.^2)-mean(y)^2

// Calcul des coefficients de la droite de régression :
a=c/v1
disp(a, "a=")
b=mean(y)-a*mean(z)
disp(b, "b=")

// Tracé de la droite de régression:
plot2d(z,a*z+b,style=3,leg='droite de regression')

// Calcul du coefficient de corrélation linéaire:
rho=c/(sqrt(v1)*sqrt(v2))
disp(rho,"rho=")

//Nuage des résidus :
scf(1), plot2d(z, y-a*z-b,style=-1)
xtitle("Nuage des résidus","z","résidus")
```

Le programme renvoit :



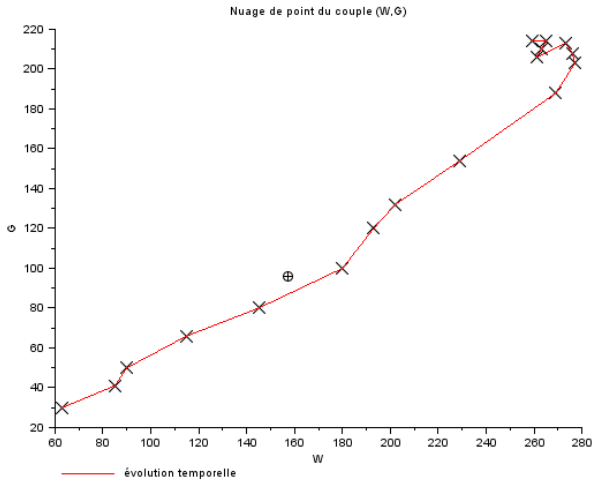



```
a=0.9909028

b=0.60890409

rho=0.9999974
```

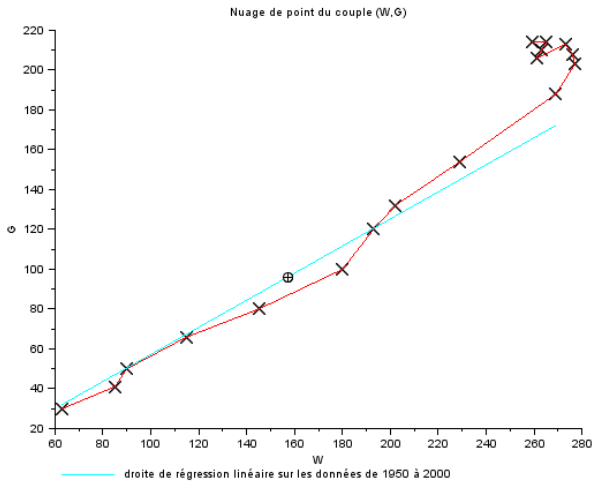
On remarque que la droite de régression linéaire, d'équation $y = ax + b$ se superpose bien au nuage de points, ce qui est confirmé par la dispersion totalement aléatoire des résidus et par un coefficient de corrélation linéaire très proche de 1.

Exercice 4.

1.
```
W=[63 85 90 115 145 180 193 202 229 269 277 276 273 261]
G=[30 41 50 66 80 100 120 132 154 188 203 208 213 206]
plot2d(W,G,style=-2)
xtitle("Nuage de point du couple (W,G)","W","G")
plot2d(W,G,style=5,leg="évolution temporelle")
```



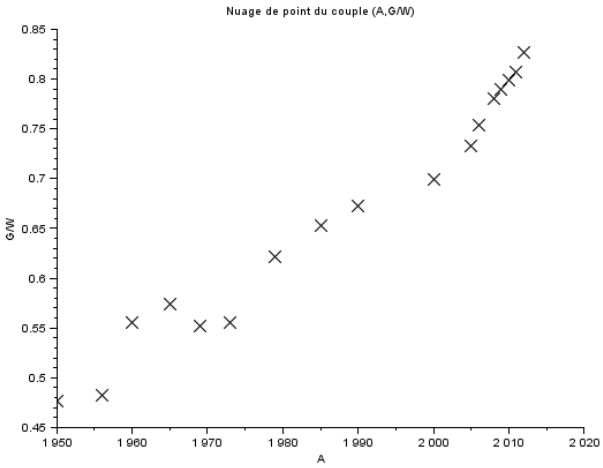
2.
```
// Etude des données entre 1950 et 2000:
W1=W(1:10)
G1=G(1:10)
plot2d(mean(W1),mean(G1),style=-3)

c=mean(W1.*G1)-mean(W1)*mean(G1)
v1= mean(W1.^2)-mean(W1)^2
v2= mean(G1.^2)-mean(G1)^2

// Droite de régression linéaire sur ces données:
a=c/v1
disp("a="+string(a))
b=mean(G1)-a*mean(W1)
disp("b="+string(b))

plot2d(W1,a*W1+b,style=4,leg="droite de régression linéaire sur les données de 1950 à 2000")
plot2d(mean(W1),mean(G1),-3)

// Coefficient de corrélation linéaire sur ces données:
rho=c/(sqrt(v1)*sqrt(v2))
disp("rho="+string(rho))
```
On obtient les résultats :




```
a=0.6805233

b=-10.810218

rho=0.8933535
```

3.
```
// Nuage de points du couple (A, G./W)
```

```
A=[1950 1956 1960 1965 1969 1973 1979 1985 1990 2000 2005 2006 2008 2009 2010 2011 2012]
scf(1)
plot2d(A, G./W, -2)
xtitle("Nuage de point du couple (A,G/W)","A","G/W")
```



Exercice 5.

1.
```
function [x,y]=simul_couple()
    x=grand(1,1,'poi',2)
    y=grand(1,1,'poi',x)
endfunction

x=zeros(1,20)
y=zeros(1,20)
for k=1:20 do
    [x(k),y(k)]=simul_couple()
end

disp([x;y])
```
Le programme crée 2 vecteurs lignes nuls x et y puis mémorise dans chaque case numéro k (k allant de 1 à 20) , une simulation du couple (X, Y) en faisant appelle à la fonction `simul_couple`.

On obtient par exemple :

```
// Couples [x;y] obtenus:
 column  1 to 17

    2.    3.    4.    2.    1.    4.    3.    5.    1.    2.    2.    1.    2.    1.    3.    4.    1.
    4.    3.    4.    2.    0.    6.    3.    4.    0.    4.    3.    1.    4.    1.    2.    1.    0.

         column 18 to 20

    0.    1.    3.
    0.    2.    5.
```

2. En faisant tourner le programme à la main, on comprend que la matrice N compte l'effectif de chaque modalité (i, j) du couple (X, Y) et donc la matrice F donne le tableau des fréquences des différentes modalités. On appelle ce tableau le tableau de contingence de la série statistique bivariée. Ici on obtient :

```
// Tableau de contingence associé à la série:
    0.05    0.      0.      0.      0.      0.      0.
    0.15    0.1     0.05    0.      0.      0.      0.
    0.      0.      0.05    0.05    0.15    0.      0.
    0.      0.      0.05    0.1     0.      0.05    0.
    0.      0.05    0.      0.      0.05    0.      0.05
    0.      0.      0.      0.      0.05    0.      0.
```

Le programme peut aussi s'écrire ainsi :

```
// Construction du tableau de contingence du couple:
l=max(X);
m=max(Y);
N=zeros(l+1,m+1)
for i=1:(l+1) do
    for j=1:m+1 do
        N(i,j)= sum((X==i-1) & (Y==j-1))
    end
end
F=N./sum(N)
disp(F)
```