```
#        Cours de Stat Mass 06: covariance
# initialisation des variables "Tailles" et "x", etc.
library(MASS)  # MASS = Modern Applied Statistics with S
data(survey)   #lecture des données de  survey
survey.cc<-survey[complete.cases(survey),] #nettoyage de  survey
Tailles<-survey.cc[, c("Wr.Hnd", "NW.Hnd", "Height")] #extraction des 3 Tailles
x<-Tailles$Wr.Hnd  #x  prends la valeur de la première colonne de  Tailles
y<-Tailles$NW.Hnd  #y  prends la valeur de la deuxième colonne de  Tailles
z<-Tailles$Height  #z  prends la valeur de la troisième colonne de  Tailles
hist(x)        #histogramme des valeurs de  x
hist(c(x,y,z))   #histogramme de la réunion (concaténation) des trois échantillons
mean(x);sd(x) # moyenne et écart-type
sd(x)^2   # le carré de l'écart-type ...
var(x)    # c'est la variance
cov(x,y)  # la covariance
cov(x,y)/(sd(x)*sd(y)) #  la formule ...
cor(x,y)  #   de la corrélation
plot(x,y) # une corrélation proche de 1 est la marque d'une alignement croissant
xx=c(min(x),max(x));xx; yy=c(min(-2*y),max(-y));yy # les bornes des dessins suivants
plot(xx,yy) # on "punaise" une figure qui recevra les dessins (points, abline,...)
points (x,-y)  # un alignement décroissant ...
cor(x,-y)   # a une corrélation proche de -1
cor(x, -2*y) # corrélation inchangée: on a juste multiplié -y par deux
points(x,-2*y,pch=2) # la pente a bien changé, mais pas la dispertion

####################### un peu de calcul mental
mean(1:3)
var(1:3) # un divise bien par (n-1)
cov(1:3,1:3) #
cor(1:3,1:3)
plot(1:3,1:3) # on a bien un alignement parfait croissant
plot(1:100,1:100) # un autre alignement parfait croissant avec beaucoup de points
cor(1:100,1:100) # la corélation est 1, comme prévisible
cov(1:100,1:100) # attention: la covariance, elle, est bien affectée par le nombre
cov(Tailles)  # la matrice de variance covariance Sigma
cor(Tailles)  # la matrice de correlation Sigma
#########################   Images et calculs
MesDonnées=data.frame(x,y,z) # on fabrique un data.frame (tableau) avec x, y, z
Reg1=lm(-y~x,data=MesDonnées);plot(x,-y)
coef(Reg1)
abline(coef(Reg1))
Reg2=lm(-2*y~x,data=MesDonnées);plot(x,-2*y)
abline(coef(Reg2))
cov(MesDonnées) # la même matrice Sigma, mais avec les noms x, y, z
cor(MesDonnées) # la matrice de corrélation
#           x         y         z
# x 1.0000000 0.9629322 0.6009909
# y 0.9629322 1.0000000 0.5841270
# z 0.6009909 0.5841270 1.0000000
plot(x,y);cor(x,y) # un nuage assez groupé cor=0.96
plot(x,z);cor(x,z) # un nuage moins groupé cor=0.60

#########################   Des données simulées:
set.seed(1515)
xSim=rnorm(200,0,1) # centrée réduite
ySim=rnorm(200,2,1) # décentrée (moyenne=2), mais réduite
zSim=rnorm(200,2,5) # décentrée et d'écart-type 5
DonnéesSimulées=data.frame(xSim,ySim,zSim) # NB: MesDonnées n'a pas changé
cov(DonnéesSimulées)
cor(DonnéesSimulées)
xx=c(min(xSim,ySim),max(xSim,ySim));yy=c(min(zSim,ySim),max(zSim,ySim))
plot(xx,yy)
points(xSim,ySim,pch=1) # les nuages sont peu organisés: la corrélation est très petite
points(xSim,zSim,pch=2)
points(ySim,zSim,pch=20)
#################      on tourne le nuage
u=xSim+zSim
v=xSim-zSim
plot(u,v)
DonnéesTournées=data.frame(u,v) # un bon regroupement ...
Reg3=lm(v~u,data=DonnéesTournées)  #autour de la droite
abline(coef(Reg3))            #de regression linéaire
cor(DonnéesTournées)  # une "forte" (-0.9053468) corrélation
```