```
#        Cours de Stat Mass 03: boxplot/quantiles/simulation
# 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
x<-survey.cc$Wr.Hnd  #x  prends la valeur de la première colonne de  survey.cc
y<-survey.cc$NW.Hnd  #y  prends la valeur de la deuxième colonne de  survey.cc
z<-survey.cc$Height  #z  prends la valeur de la troisième colonne de  survey.cc
# une fonction bien utile: sort()
sort(x)
hist(x)        ###   histogramme des valeurs de  x
################ On refait le "même" histogramme avec une syntaxe choisissant les bornes
xmin<- floor(min(x))-1;xmax<- ceiling(max(x))
xmin;xmax #plus petite et plus grande abscisse qui seront affichées par "hist"
bornes=xmin:xmax;bornes
hist(x,breaks=bornes,col="green",add=T) # notez que la classe inférieure est dédoublée
####### la plus petite borne inférieurs est comprise!
xmin<- floor(min(x));xmax<- ceiling(max(x))
xmin;xmax #plus petite et plus grande abscisse qui seront affichées par "hist"
bornes=xmin:xmax;bornes
hist(x,breaks=bornes,col="yellow",add=T) # notez que la classe inférieure est dédoublée
####### un nouveau "résumé graphique": la boîte-à-moustache (whiskers plot de Tukey 1977)
     # fenêtres graphiques multiples:    (en savoir plus: ?device)
     dev.list()
     dev.cur()
     windows()   # on aurait aussi pu ouvrir cette fenêtre avec  x11()
boxplot(x)
boxplot(y)
     dev.set(2);hist(y) #on revient d'abord à la fenêtre comportant l'histogramme de x
sort(y)   # ici on peut voir les "outliers" 12.5 13.0 13.5 de y
boxplot(x,y,z)  # les trois boites-à-moustaches côte à côte
hist(x,breaks=bornes,col="green") # revenons à  x
dev.set(3);boxplot(x);x.sort=sort(x);x.sort
n=length(x);n;x.sort[n/4];x.sort[n/2];x.sort[n*3/4]
mu=mean(x);mu
sigma=sd(x);sigma
qnorm(1/4,mu,sigma);qnorm(1/2,mu,sigma);qnorm(3/4,mu,sigma);
# dnorm(x,mu,sigma) donne la densité f(x) de la loi normale,##Rappels sur les fonction normales
# pnorm(x,mu,sigma) donne la fonction de répartition F(x) de la loi normale,
# qnorm(p,mu,sigma) donne le quantile d'un probabilité p de la loi normale , et
# rnorm(n,mu,sigma) génère un n-échantillon suivant la loi normale N(mu,sigma).
######################" comparaison avec une loi "normale" de même caractéristiques
# recommençons avec un échantillon "normal" de même caractéristiques
x.norm=rnorm(n,mu,sigma)
x.sorm=sort(x.norm);x.sorm    #le même, trié. Ici "sorm" pour "sort.norm"
windows();windows();    # on ouvre deux nouvelles fenêtres
dev.set(4);hist(x.norm) # on choisi la fenêtre 4
dev.set(5);boxplot(x.norm)
boxplot(x,x.norm)
grosx.norm=rnorm(n*100,mu,sigma)
boxplot(x,grosx.norm)
x.sort    # ici on peut voir l'"outliers" 13.0 de x
# les quartiles Q1, Q2, et Q3, et extrema des moustaches (x_i "adjacents)
Q1=function(x){return(sort(x)[length(x)/4])}
Q1(x)
Q2=function(x){return(sort(x)[length(x)/2])}
Q2(x)
Q3=function(x){return(sort(x)[length(x)*3/4])}
Q3(x)
# Les moustaches! (whyskers)
MaxMoustache=function(x){return(Q3(x)+1.5*(Q3(x)-Q1(x)))}
MaxMoustache(x)
MinMoustache=function(x){return(Q1(x)-1.5*(Q3(x)-Q1(x)))}
MinMoustache(x)
boxplot(x,x.norm)
x.sort #observez les "exceptionnels" (outliers)
x.sorm  # idem sur un echantillon "normal"
MaxMoustache(x.sorm)
MinMoustache(x.sorm)
# quartiles normaux
pnorm(-1)
Q1.norm=qnorm(1/4);Q1.norm
Q2.norm=qnorm(0.5);Q2.norm
Q3.norm=qnorm(3/4);Q3.norm
```