```
#        Cours de Stat Mass 04
#                Quantiles
rnorm(1,0,1) # vérification que cela commence toujours pareil ou non
set.seed(2015) # on s'assure que cela commencera toujours pareil
###################### quelques fonctions liées à la loi normale
# dnorm(x, mean = 0, sd = 1, log = FALSE)
# pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
# qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
# rnorm(n, mean = 0, sd = 1)
#################################################################
# initialisation des variables "Tailles" et "x"
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  #x  prends la valeur de la première colonne de  Tailles.cc
hist(x)        #histogramme des valeurs de  x
hist(x,freq=F)
n=length(x);n

################# quartile ############################
d=4
d;n/d  # affichage du fractile et de la taille (idéale) des classes
(1:d)*n/d   # utile pour les indices de quartiles
indicesbornesx=floor((1:d)*n/d);indicesbornesx # on arrondi à l'entier inférieur
sort(x)   #affichage trié
bornesx=1:(d+1);bornesx  #creation du vecteur
bornesx[1]=sort(x)[1];bornesx[1]   #on commence par le plus petit
bornesx[2:(d+1)]=sort(x)[indicesbornesx]
bornesx  # affichage des quantiles et des élements extrèmes
hist(x,breaks=bornesx)
hist(x,breaks=bornesx,freq=T) #

################  deciles ######################################
d=10
d;n/d  # affichage du fractile et de la taille (idéale) des classes
(1:d)*n/d
indicesbornesx=floor((1:d)*n/d);indicesbornesx # on arrondi à l'entier inférieur
sort(x)   #affichage trié
bornesx=1:(d+1);bornesx  #creation du vecteur
bornesx[1]=sort(x)[1];bornesx[1]   #on commence par le plus petit
bornesx[2:(d+1)]=sort(x)[indicesbornesx]
bornesx  # affichage des quantiles et des élements extrèmes
hist(x,breaks=bornesx)
hist(x,breaks=bornesx,freq=T) # il y a énormément de 17.5: ils vont tous dans la classe ]16.5,17.
hist(x,breaks=bornesx)  # retour à l'histogramme en densité

######################" comparaison avec une loi "normale" de même caractéristiques
mu=mean(x);mu
sigma=sd(x);sigma  # sd vient de "standard deviation"
magaussienne=function(x){return(dnorm(x,mu,sigma))}
xmin=min(x);xmax=max(x)
plot(magaussienne,xmin,xmax,col="red",add=TRUE) # tracé du graphe de la fonction entre  xmin et x

#################   deciles théoriques
d=10
bornesthx=qnorm((0:d)/d,mu,sigma);bornesthx
bornesthxBis=bornesthx
bornesthxBis[1]=xmin # on triche: on remplace -Inf par xmin
bornesthxBis[11]=xmax # on triche: on remplace Inf par xmax
hist(x,breaks=bornesthxBis,col="red",add=T) #
hist(x,breaks=bornesx,col="yellow",add=T)
bornesx;bornesthx
qqplot(bornesx,bornesthx)
x.norm=rnorm(length(x),mean(x),sd(x))
qqplot(x.norm,x)

###################  Quartiles théorique
boxplot(x)
mu=mean(x);mu
sigma=sd(x);sigma  # sd vient de "standard deviation"
Q=qnorm(0.25*(1:3),mu,sigma);Q
Q[3]-Q[1] # interquartile
L=1.5*(Q[3]-Q[1]);L # longueur maximale des moustache:
sort(x) #affichage trié
```

```
# Tout caractère inférieur qmin ou supérieur à qmax est exceptionnel (outliers)
qmin=Q[1]-L;qmin
qmax=Q[3]+L;qmax
pmin=pnorm(qmin,mu,sigma);pmin # proba théorique d'être inférieur à qmin
# déciles théorique
D=qnorm(0.1*(1:9),mu,sigma);D
d=10
indicesbornesx=floor((1:d)*n/d);indicesbornesx # c'est la première classe qui est trop petite

###################### échantillons gaussiens centré réduit
set.seed(1515)  # choix de la "graine" d'initiation des nombres aléatoires
n=1000
u=rnorm(n,0,1)
hist(u)
h=hist(u,freq=F)
magaussienne=function(x){return(dnorm(x,0,1))}
plot(magaussienne,min(u),max(u),col="red",add=T) # tracé du graphe de la fonction entre  xmin et
plot(h,add=T,freq=F)
d=4;(0:d)/d
u.sort=sort(u);u.sort
bornes=1:(d+1);bornes # creation de d; sinon "objet 'bornes' introuvable" à la suivante
bornes[1]=u.sort[1];bornes[1]
(1:d)*n/d
u.sort[(1:d)*n/d]
bornes[2:(d+1)]=u.sort[(1:d)*n/d]
bornes
h=hist(u,breaks=bornes,freq=T)
h$xname <- "u normal (effectifs)"
plot(h,freq=T)
hist(u,breaks=bornes,freq=F)
hist(u,breaks=bornes)
magaussienne=function(x){return(dnorm(x,0,1))}
plot(magaussienne,u.sort[1],u.sort[n],col="red",add=TRUE) # tracé du graphe de la fonction entre
```