```
#        Cours de Stat Mass 01
library(MASS)  #Chargement de la bibliothèque de codes MASS=
#                    Modern Applied Statistics with S
#  data(survey)   #lecture des données de  survey ???
names(survey)  #les noms des caractères de  survey
head(survey)   #les premières lignes de  survey
fix(survey)    #un tableur permettant de lire et modifier les valeurs de  survey
survey.cc <- survey[complete.cases(survey), ]  #nettoyage des données
Tailles<-survey[, c("Wr.Hnd", "NW.Hnd", "Height")] #extraction des trois colonnes
Tailles #affichage des  Tailles
Tailles.cc<-Tailles[complete.cases(Tailles),] #nettoyage de  Tailles
Tailles.cc  #affichage de Tailles.cc
x<-Tailles.cc$Wr.Hnd  #x  prends la valeur de la première colonne de  Tailles.cc
fix(x)   #attention: ne changez rien; vous pouvez regarder ce que sont les lignes de x
hist(x)->h  #histogramme des valeurs de  x
h$counts <- cumsum(h$counts) # replace the cell freq.s by cumulative freq.s
plot( h ) # plot a cumulative histogram of x
hist(x, breaks=c(min(x),20,22,23,max(x)))
hist(x,breaks=seq(13,25,2))
hist(x,freq=FALSE)
xmin<-min(x);xmax<-max(x) #plus petite et plus grande valeur de  x
hist(x,xlim=c(xmin,xmax))#une syntaxe qui pourrait permettre de changer l'affichage
hist(x,add=TRUE) #une syntaxe qui permet d'ajouter l'histogramme à l'existant

######################" un échantillon "normal" de même caractéristiques
mu=mean(x) #moyenne des valeurs de  x   #notez qu'on peut remplacer  <-  par  =
sigma=sd(x)  #écart-type des valeurs de  x
longueur=length(x)  #nombre d'éléments de  x
y=rnorm(longueur,mu,sigma) #génération aléatoire d'un caractère normal y ayant même mu et sigma
hist(y,add=TRUE) "superposition de l'histogramme de  y

###################### 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)
# dnorm gives the density,
# pnorm gives the distribution function,
# qnorm gives the quantile function, and
# rnorm generates random deviates.

magaussienne=function(x){return(longueur*dnorm(x,mu,sigma))} #exemple de déclaration de fonction
par(new=TRUE) #passe magique pour ajouter un dession à la fen^tre courante
plot(magaussienne,xmin,xmax) # tracé du graphe de la fonction entre  xmin et xmax
################## On recommence avec un échantillon "normal" de taille "Longueur" variable
Longueur=longueur*100 # on commence avec cent fois plus de données
y=rnorm(Longueur,mu,sigma) #     --"--
ymin=min(y)-5;ymax=max(y)+5 #on se prépare à afficher entre des valeurs plus espacées
hist(y) # histogramme usuel
hist(y,xlim=c(ymin,ymax))# histogramme élargi
MaGaussienne=function(x){return(Longueur*dnorm(x,mu,sigma))}
par(new=TRUE) #passe magique pour ajouter un dessin à la fenêtre courante
plot(MaGaussienne,ymin,ymax) # tracé du modèle gaussien pour ces données
################## règlage de la hauteur du graphe
hauteur=MaGaussienne(mu) # la plus grande valeur de MaGaussienne à afficher
plot(MaGaussienne,ymin,ymax)
par(new=TRUE)
hist(y,xlim=c(ymin,ymax),ylim=c(0,hauteur)) #la fonction  c  retourne la "liste" de ses vale

################## affichage et coloriage des données
survey.cc <- survey[complete.cases(survey), ]
#Nous récupérons le sexe des individus dans un vecteur, et définissons un
#vecteur de couleurs pour distinguer facilement les filles des garcons :
sexe <- survey.cc$Sex
couleur <- ifelse(sexe == "Male", "blue", "red")
#Nous extrayons trois variables quantitatives de ce jeu de données : l'empan de
#la main d'écriture dite main dominante, l'empan de la main non dominante et
#la stature. Toutes sont exprimées en centimètres.
mesures <- survey.cc[, c("Wr.Hnd", "NW.Hnd", "Height")]
names(mesures) <- c("Empan1", "Empan2", "Taille") #traduction des noms des caractères
head(mesures)
plot(mesures, col = couleur, pch = 19)
#Notez que les variables morphométriques des garçons (en bleu) sont en moyenne
#plus grandes que celles des filles (en rouge). Ceci nous permettra de nous orienter
```

```
#plus facilement dans les representations graphiques suivantes.
#Chaque individu est caractérisé par trois variables, soit par un point dans R3. La
#fonction plot3d() de la bibliothèque rgl [1] vous permet d'explorer facilement
#un nuage de points en 3 dimensions. Mais ceci demande à charger la bibliothèque
# rgl et c'est une autre histoire ;-) #library(rgl)
```