L'histogramme

- Outil important de statistique descriptive.

L'histogramme

- Outil important de statistique descriptive.
- Données $x_1, \ldots, x_n$, $\quad b_0 \leq \min_i x_i \leq \max_i x_i \leq b_k$

L'histogramme

- Outil important de statistique descriptive.
- Données $x_1, \ldots, x_n$, $\quad b_0 \leq \min_i x_i \leq \max_i x_i \leq b_k$
- $[b_0, b_1], (b_1, b_2], \ldots, \ldots, (b_{k-1}, b_k]$, $\qquad b_0$: point d"ancrage

L'histogramme

- Outil important de statistique descriptive.
- Données $x_1, \ldots, x_n$, $\quad b_0 \leq \min_i x_i \leq \max_i x_i \leq b_k$
- $[b_0, b_1], (b_1, b_2], \ldots, \ldots, (b_{k-1}, b_k]$, $\qquad b_0$: point d''ancrage
- Paramètre de lissage: $h = b_j - b_{j-1}$

L'histogramme

- Outil important de statistique descriptive.
- Données $x_1, \ldots, x_n$, $\quad b_0 \leq \min_i x_i \leq \max_i x_i \leq b_k$
- $[b_0, b_1], (b_1, b_2], \ldots, \ldots, (b_{k-1}, b_k]$, $\qquad b_0$: point d"ancrage
- Paramètre de lissage: $h = b_j - b_{j-1}$
- $n_j = \#\{x_i \in (b_{j-1}, b_j]\}$

$$\begin{aligned}
\widehat{f}(x) &= \frac{\widehat{F}(b_{j+1}) - \widehat{F}(b_j)}{h} \\
&= \frac{(\#\{x_i \leq b_{j+1}\} - \#\{x_i \leq b_j\})/n}{h} \\
&= \frac{n_j}{nh}, \qquad x \in (b_j, b_{j+1}].
\end{aligned}$$

L'histogramme

- Outil important de statistique descriptive.
- Données $x_1, \ldots, x_n$, $\quad b_0 \leq \min_i x_i \leq \max_i x_i \leq b_k$
- $[b_0, b_1], (b_1, b_2], \ldots, \ldots, (b_{k-1}, b_k]$, $\qquad b_0$: point d"ancrage
- Paramètre de lissage: $h = b_j - b_{j-1}$
- $n_j = \#\{x_i \in (b_{j-1}, b_j]\}$

$$\begin{aligned}
\widehat{f}(x) &= \frac{\widehat{F}(b_{j+1}) - \widehat{F}(b_j)}{h} \\
&= \frac{(\#\{x_i \leq b_{j+1}\} - \#\{x_i \leq b_j\})/n}{h} \\
&= \frac{n_j}{nh}, \qquad x \in (b_j, b_{j+1}].
\end{aligned}$$

- Aire sous l"histogramme $= \sum_j \frac{n_j}{nh} \times h = 1$

L’histogramme dans R

- La fonction `hist`
  hist(x, breaks = "Sturges", freq = NULL, probability = !freq, include.lowest = TRUE, right = TRUE, density = NULL, angle = 45, col = NULL, border = NULL, main = paste("Histogram of" , xname), xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE, plot = TRUE, labels = FALSE, nclass = NULL, ...)

L’histogramme dans R

- La fonction `hist`
  hist(x, breaks = "Sturges", freq = NULL, probability = !freq, include.lowest = TRUE, right = TRUE, density = NULL, angle = 45, col = NULL, border = NULL, main = paste("Histogram of" , xname), xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE, plot = TRUE, labels = FALSE, nclass = NULL, ...)
- Par défaut: 1) intervalles de même longueur; 2) règle de Sturges; 3) fréquences en ordonnée.

L'histogramme dans R

- La fonction `hist`
  hist(x, breaks = "Sturges", freq = NULL, probability = !freq, include.lowest = TRUE, right = TRUE, density = NULL, angle = 45, col = NULL, border = NULL, main = paste("Histogram of" , xname), xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE, plot = TRUE, labels = FALSE, nclass = NULL, ...)
- Par défaut: 1) intervalles de même longueur; 2) règle de Sturges; 3) fréquences en ordonnée.
- `breaks`: règle "Sturges", "Scott" ou "FD". Autres valeurs possibles: n. d'intervalles voulu (succès non garanti), le vecteur des bornes des intervalles (garanti).

L'histogramme dans R

- La fonction `hist`
  hist(x, breaks = "Sturges", freq = NULL, probability = !freq, include.lowest = TRUE, right = TRUE, density = NULL, angle = 45, col = NULL, border = NULL, main = paste("Histogram of" , xname), xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE, plot = TRUE, labels = FALSE, nclass = NULL, ...)
- Par défaut: 1) intervalles de même longueur; 2) règle de Sturges; 3) fréquences en ordonnée.
- `breaks`: règle "Sturges", "Scott" ou "FD". Autres valeurs possibles: n. d'intervalles voulu (succès non garanti), le vecteur des bornes des intervalles (garanti).
- Les arguments `freq=FALSE` ou `probability=TRUE` produisent un véritable estimateur de densité.

L’histogramme dans R

- Au lieu d’un graphique, la commande
  > hist(x,plot = FALSE)
  produit une liste à 7 composantes

L'histogramme dans R

- Au lieu d'un graphique, la commande
  > hist(x,plot = FALSE)
  produit une liste à 7 composantes
- Les composantes les plus importantes sont:
  $breaks # les bornes des intervalles
  $counts # les fréquences de chacun des intervalles
  $density # les valeurs des $\widehat{f}(b_j)$
  $mids # les points milieux des intervalles

L'histogramme dans R

- Au lieu d'un graphique, la commande
  > hist(x,plot = FALSE)
  produit une liste à 7 composantes
- Les composantes les plus importantes sont:
  $breaks # les bornes des intervalles
  $counts # les fréquences de chacun des intervalles
  $density # les valeurs des $\widehat{f}(b_j)$
  $mids # les points milieux des intervalles
- On peut aussi construire un histogramme à partir de la fonction `truehist` du package `MASS`. Celle-ci produit par défaut un véritable estimateur de densité.

L'histogramme dans R

- Au lieu d'un graphique, la commande
  > hist(x,plot = FALSE)
  produit une liste à 7 composantes
- Les composantes les plus importantes sont:
  $breaks # les bornes des intervalles
  $counts # les fréquences de chacun des intervalles
  $density # les valeurs des $\widehat{f}(b_j)$
  $mids # les points milieux des intervalles
- On peut aussi construire un histogramme à partir de la fonction `truehist` du package `MASS`. Celle-ci produit par défaut un véritable estimateur de densité.
- `truehist` ignore la règle de Sturges, mais propose celles de Scott et Freedman-Diaconis. Arguments: `nbins` (n. d'intervalles) et h (paramètre de lissage).

Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes

Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes
- > x = read.table("swissmon.dat")[,1] #100 observations de la variable V1
  #V1 largeur en mm de la marge d''un billet contrefait

Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes
- > x = read.table("swissmon.dat")[,1] #100 observations de la variable V1
  #V1 largeur en mm de la marge d''un billet contrefait
- > par(mfrow=c(1,3)) #3 graphiques sur 1 ligne

Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes
- > x = read.table("swissmon.dat")[,1] #100 observations de la variable V1
  #V1 largeur en mm de la marge d''un billet contrefait
- > par(mfrow=c(1,3)) #3 graphiques sur 1 ligne
- > hist(x,breaks=28,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="blue")
  > hist(x,breaks=12,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="red")
  > hist(x,breaks=6,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="orange")

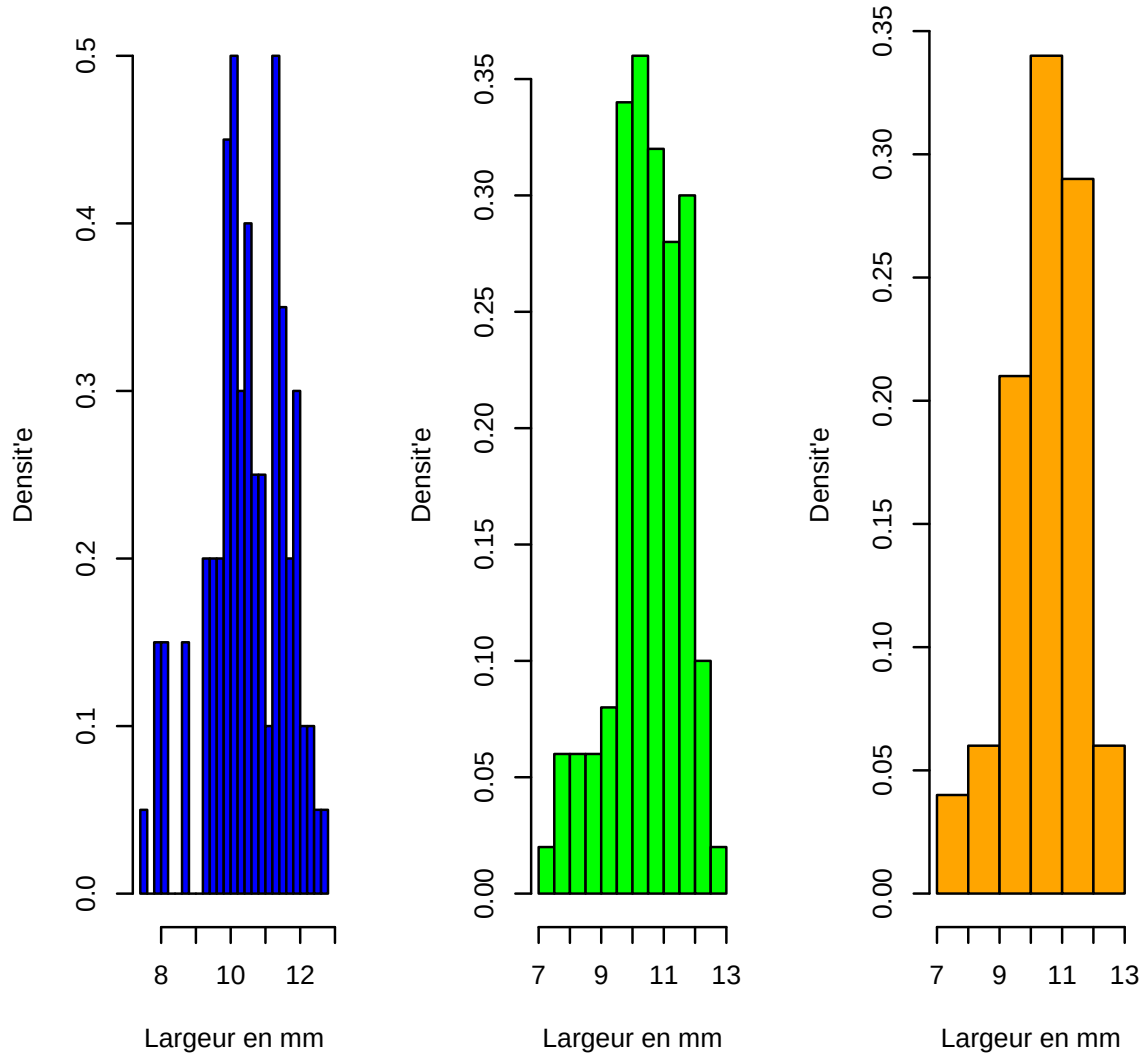
Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes
- > x = read.table("swissmon.dat")[,1] #100 observations de la variable V1
  #V1 largeur en mm de la marge d"un billet contrefait
- > par(mfrow=c(1,3)) #3 graphiques sur 1 ligne
- > hist(x,breaks=28,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="blue")
  > hist(x,breaks=12,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="red")
  > hist(x,breaks=6,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="orange")
- Différents lots de billets contrefaits $\Longrightarrow$ multimodalité

Effet du paramètre de lissage

- Jeu `swissmon.dat` #tableau à 4 var., 100 lignes
- > x = read.table("swissmon.dat")[,1] #100 observations de la variable V1
  #V1 largeur en mm de la marge d"un billet contrefait
- > par(mfrow=c(1,3)) #3 graphiques sur 1 ligne
- > hist(x,breaks=28,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="blue")
  > hist(x,breaks=12,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="red")
  > hist(x,breaks=6,xlab="Largeur en mm",ylab="Densité",main="",freq=F,col="orange")
- Différents lots de billets contrefaits $\Longrightarrow$ multimodalité
- Premier histogramme: multimodal (sous-lissage); deuxième: 2 modes; troisième: 1 mode (surlissage).

swissmon.dat: 28, 12 et 6 classes



Détermination du n. de classes

- Seule façon garantie d'obtenir le n. d'intervalles désiré:
  breaks = vecteur des bornes des intervalles

Détermination du n. de classes

- Seule façon garantie d'obtenir le n. d'intervalles désiré:
  breaks = vecteur des bornes des intervalles
- nombre de classes de la méthode de Sturges
  > nclass.Sturges(x) #x ci-dessus
  [1] 8

Détermination du n. de classes

- Seule façon garantie d'obtenir le n. d'intervalles désiré:
  breaks = vecteur des bornes des intervalles
- nombre de classes de la méthode de Sturges
  > nclass.Sturges(x) #x ci-dessus
  [1] 8
- nombre de classes de la méthode de Scott
  > nclass.scott(x)
  [1] 7

Détermination du n. de classes

- Seule façon garantie d'obtenir le n. d'intervalles désiré:
  breaks = vecteur des bornes des intervalles
- nombre de classes de la méthode de Sturges
  > nclass.Sturges(x) #x ci-dessus
  [1] 8
- nombre de classes de la méthode de Scott
  > nclass.scott(x)
  [1] 7
- nombre de classes de la méthode de Friedman-Diaconis
  > nclass.FD(x)
  [1] 9

Détermination du n. de classes

- Seule façon garantie d'obtenir le n. d'intervalles désiré:
  breaks = vecteur des bornes des intervalles
- nombre de classes de la méthode de Sturges
  > nclass.Sturges(x) #x ci-dessus
  [1] 8
- nombre de classes de la méthode de Scott
  > nclass.scott(x)
  [1] 7
- nombre de classes de la méthode de Friedman-Diaconis
  > nclass.FD(x)
  [1] 9
- La règle de Sturges est déconseillée lorsque $n > 200$.

Défaillance de la règle de Sturges

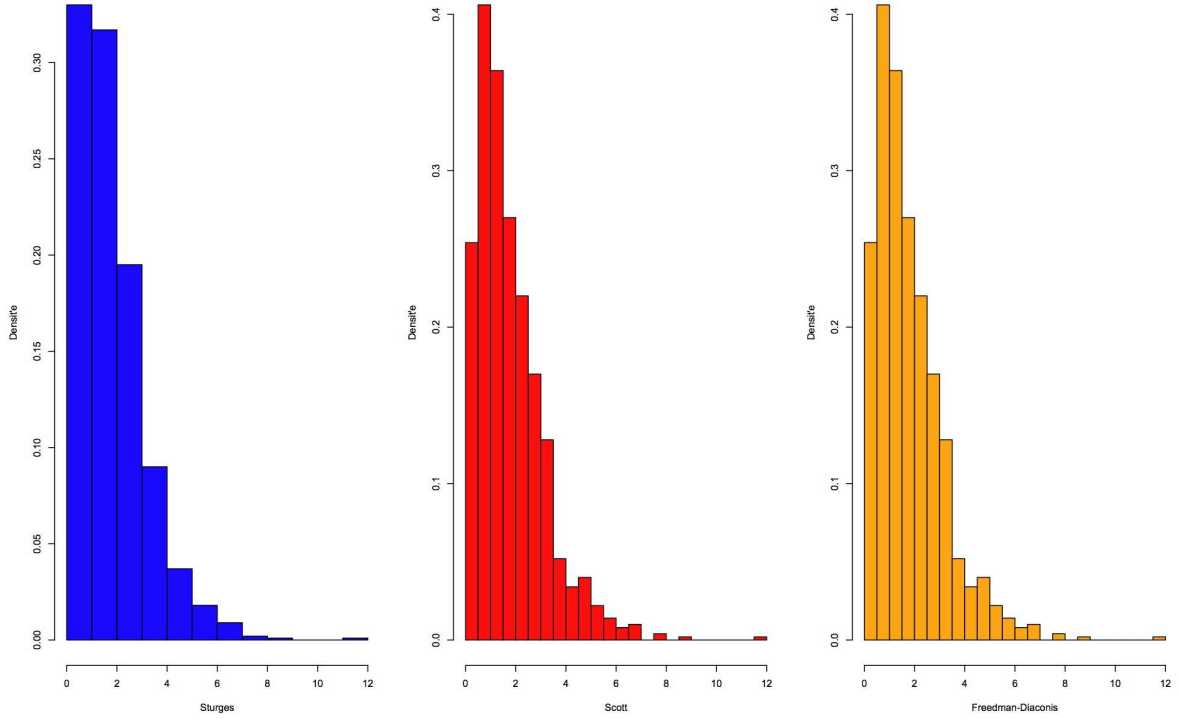
- > set.seed(129); x = rgamma(1000, 1.75)

Défaillance de la règle de Sturges

- > set.seed(129); x = rgamma(1000, 1.75)
- Les n. de classes des 3 règles sont:
  11 (Sturges), 25 (Scott), 36 (Freedman-Diaconis)

Défaillance de la règle de Sturges

- > set.seed(129); x = rgamma(1000, 1.75)
- Les n. de classes des 3 règles sont:
  11 (Sturges), 25 (Scott), 36 (Freedman-Diaconis)
- > par(mfrow=c(1,3) #3 histogrammes en parallèle



Choix du point d'ancrage

- Loi mélange: $0.5 * N(0,1) + 0.5 * N(3,1)$

$$X = \begin{cases} N(0,1) & \text{avec proba. } 0.5 \\ N(3,1) & \text{avec proba. } 0.5 \end{cases}$$

Choix du point d'ancrage

- Loi mélange: $0.5 * N(0,1) + 0.5 * N(3,1)$

$$X = \begin{cases} N(0,1) & \text{avec proba. } 0.5 \\ N(3,1) & \text{avec proba. } 0.5 \end{cases}$$

- densité de X

$$f(x) = 0.5 \times \phi(x) + 0.5 \times \phi(x-3), \quad -\infty < x < \infty$$

où $\phi(x) = \exp(-x^2/2)/\sqrt{2\pi}$.

Choix du point d'ancrage

- Loi mélange: $0.5 * N(0,1) + 0.5 * N(3,1)$

$$X = \begin{cases} N(0,1) & \text{avec proba. } 0.5 \\ N(3,1) & \text{avec proba. } 0.5 \end{cases}$$

- densité de X

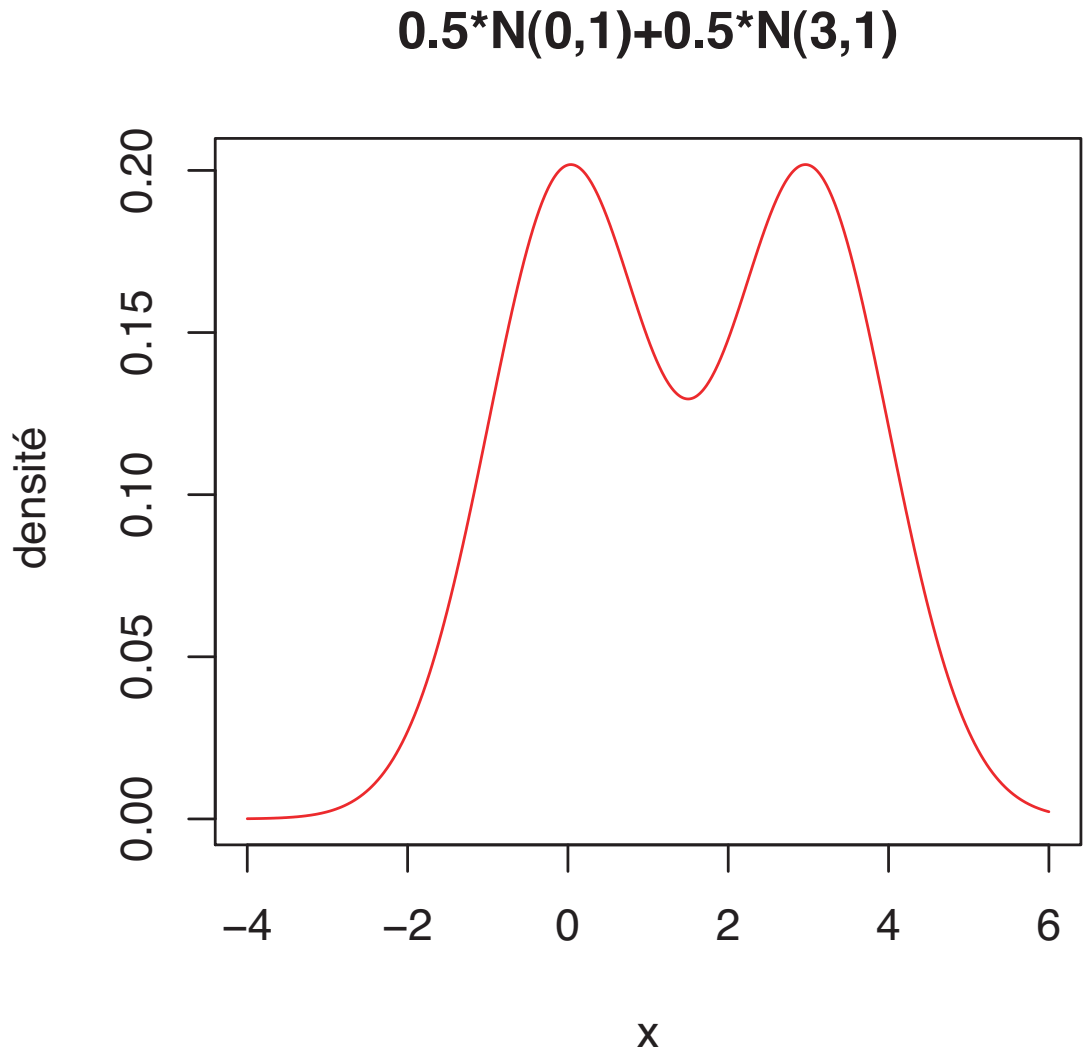
$$f(x) = 0.5 \times \phi(x) + 0.5 \times \phi(x-3), \quad -\infty < x < \infty$$

où $\phi(x) = \exp(-x^2/2)/\sqrt{2\pi}$.

- Graphique

```
> a = seq(-4,6,.02)
> plot(a,0.5*dnorm(a)+0.5*dnorm(a,3,1),ylab="densité",
main = "0.5*N(0,1)+0.5*N(3,)",col="red",type="l")
```

Graphe de la densité du mélange



Choix du point d'ancrage (suite)

- Simulation de 50 obs. de $0.5 * N(0,1) + 0.5 * N(3,1)$
  > y = numeric(50); set.seed(399)
  > for (i in 1:50){if (runif(1)<=0.5) y[i] = rnorm(1)
  else y[i] = rnorm(1,mean=3,sd=1)}

Choix du point d'ancrage (suite)

- Simulation de 50 obs. de $0.5 * N(0,1) + 0.5 * N(3,1)$
  > y = numeric(50); set.seed(399)
  > for (i in 1:50){if (runif(1)<=0.5) y[i] = rnorm(1)
  else y[i] = rnorm(1,mean=3,sd=1)}
- > range(y) # donne le min et le max
  [1] -2.082688 4.920506

Choix du point d'ancrage (suite)

- Simulation de 50 obs. de $0.5 * N(0,1) + 0.5 * N(3,1)$
  > y = numeric(50); set.seed(399)
  > for (i in 1:50){if (runif(1)<=0.5) y[i] = rnorm(1)
  else y[i] = rnorm(1,mean=3,sd=1)}
- > range(y) # donne le min et le max
  [1] -2.082688 4.920506
- Posons
  > a = c(-2.1, -1.1, -0.1, 0.9, 1.9, 2.9, 3.9, 4.9,5.9)
  # point d'ancrage -2.1, 8 intervalles de longueur 1

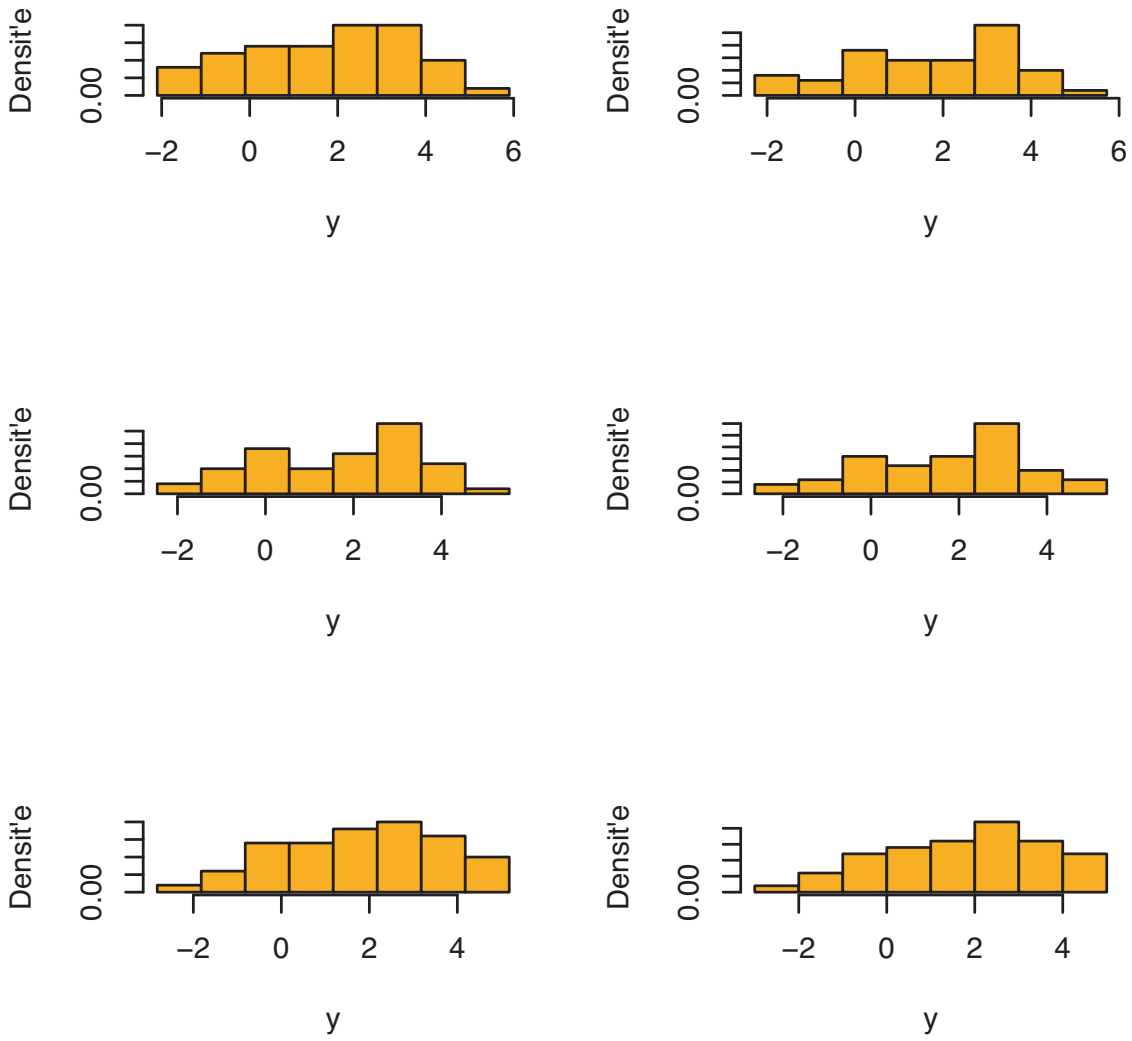
Choix du point d'ancrage (suite)

- Simulation de 50 obs. de $0.5 * N(0,1) + 0.5 * N(3,1)$
  > y = numeric(50); set.seed(399)
  > for (i in 1:50){if (runif(1)<=0.5) y[i] = rnorm(1)
  else y[i] = rnorm(1,mean=3,sd=1)}
- > range(y) # donne le min et le max
  [1] -2.082688 4.920506
- Posons
  > a = c(-2.1, -1.1, -0.1, 0.9, 1.9, 2.9, 3.9, 4.9,5.9)
  # point d'ancrage -2.1, 8 intervalles de longueur 1
- par(mfrow=c(3,2)) #3 lignes de 2 graphiques chacune

Choix du point d'ancrage (suite)

- Simulation de 50 obs. de $0.5 * N(0,1) + 0.5 * N(3,1)$
  > y = numeric(50); set.seed(399)
  > for (i in 1:50){if (runif(1)<=0.5) y[i] = rnorm(1)
  else y[i] = rnorm(1,mean=3,sd=1)}
- > range(y) # donne le min et le max
  [1] -2.082688 4.920506
- Posons
  > a = c(-2.1, -1.1, -0.1, 0.9, 1.9, 2.9, 3.9, 4.9,5.9)
  # point d'ancrage -2.1, 8 intervalles de longueur 1
- par(mfrow=c(3,2)) #3 lignes de 2 graphiques chacune
- > for (i in seq(0,-0.9,by=.15)
  hist(y,breaks=a+i,probability=T,ylab="Densité",
  main="",col="blue") # 6 histogrammes

Choix du point d'ancrage (suite)



Polygone de fréquence

- > taux = read.table("cdrate.dat")[,1]#69 taux d"intérêt
  > ht=hist(taux,xlim=c(7.3,8.9),ylim=c(0,1.4),xlab="taux", ylab="densité",probability=T,main="Polygone de fréquence",col="orange")

Polygone de fréquence

- > taux = read.table("cdrate.dat")[,1]#69 taux d"intérêt
  > ht=hist(taux,xlim=c(7.3,8.9),ylim=c(0,1.4),xlab="taux", ylab="densité",probability=T,main="Polygone de fréquence",col="orange")
- > ht$mids #7.5 7.7 7.9 8.1 8.3 8.5 8.7 (7 points milieux)

Polygone de fréquence

- > taux = read.table("cdrate.dat")[,1]#69 taux d"intérêt
  > ht=hist(taux,xlim=c(7.3,8.9),ylim=c(0,1.4),xlab="taux", ylab="densité",probability=T,main="Polygone de fréquence",col="orange")
- > ht$mids #7.5 7.7 7.9 8.1 8.3 8.5 8.7 (7 points milieux)
- Nous allons superposer le polygone à l'histogramme.
  > par(new=T) #commande la superposition

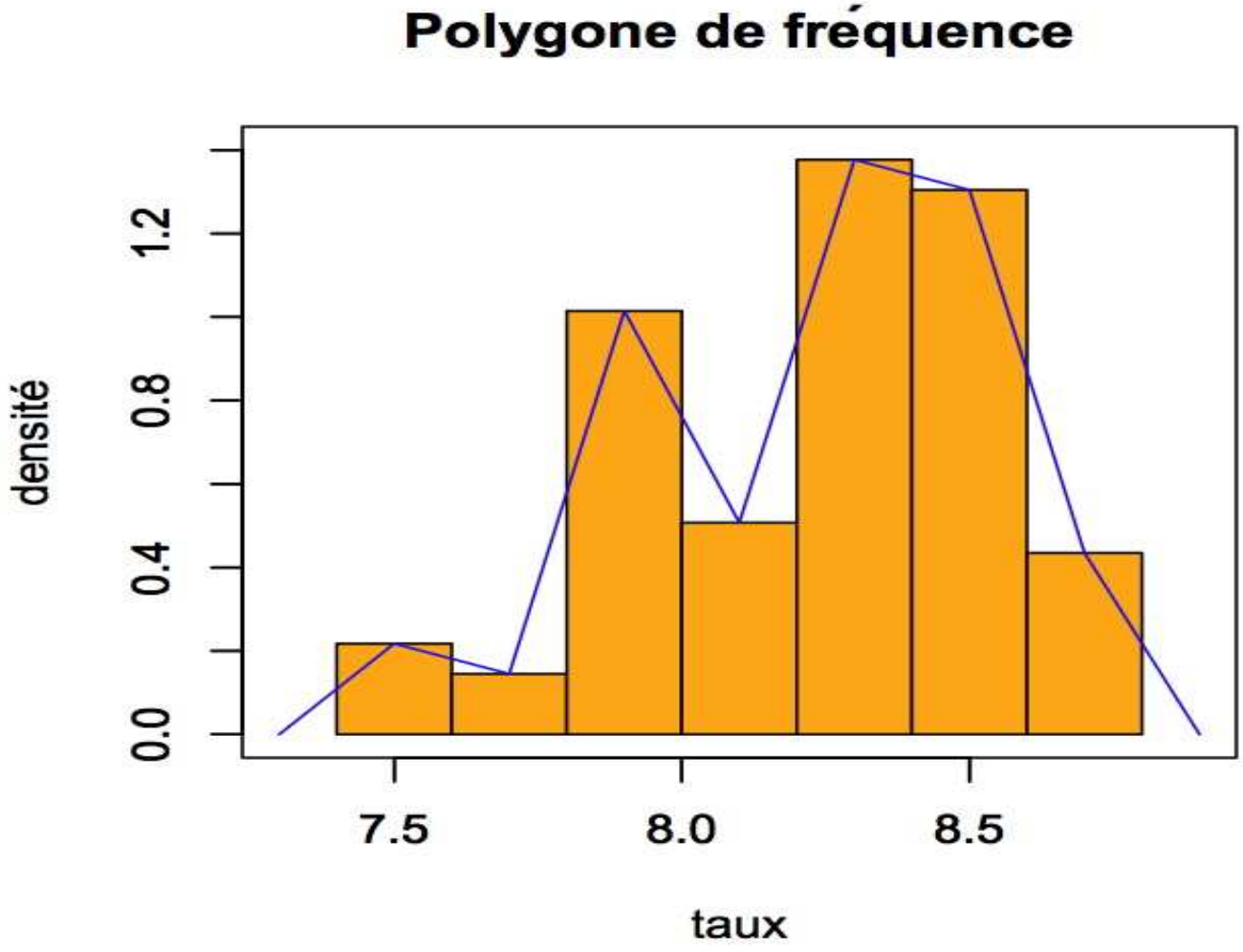
Polygone de fréquence

- > taux = read.table("cdrate.dat")[,1]#69 taux d"intérêt
  > ht=hist(taux,xlim=c(7.3,8.9),ylim=c(0,1.4),xlab="taux", ylab="densité",probability=T,main="Polygone de fréquence",col="orange")
- > ht$mids #7.5 7.7 7.9 8.1 8.3 8.5 8.7 (7 points milieux)
- Nous allons superposer le polygone à l'histogramme.
  > par(new=T) #commande la superposition
- > a = c(7.3,ht$mids,8.9) #9 points milieux
  > b = c(0,ht$density,0) #9 ordonnées
  > plot(a,b,type="l",xlim=c(7.3,8.9),ylim=c(0,1.4), col="blue"); abline(h=0)

Polygone de fréquence

- > taux = read.table("cdrate.dat")[,1]#69 taux d"intérêt
  > ht=hist(taux,xlim=c(7.3,8.9),ylim=c(0,1.4),xlab="taux", ylab="densité",probability=T,main="Polygone de fréquence",col="orange")
- > ht$mids #7.5 7.7 7.9 8.1 8.3 8.5 8.7 (7 points milieux)
- Nous allons superposer le polygone à l'histogramme.
  > par(new=T) #commande la superposition
- > a = c(7.3,ht$mids,8.9) #9 points milieux
  > b = c(0,ht$density,0) #9 ordonnées
  > plot(a,b,type="l",xlim=c(7.3,8.9),ylim=c(0,1.4), col="blue"); abline(h=0)
- Pour une superposition parfaite, il est essentiel d'inclure les mêmes `xlim, ylim` dans les 2 commandes.

Polygone de fréquence (suite)



Polygone de fréquence (suite)

- La fonction `freq.poly`

```
freq.poly function(x, br=NULL ) {
if(missing(br)) br = hist(x, plot = F)$breaks
fr = hist(x, , br, plot = F)$counts
h=br[2]-br[1]; n = length(fr)
centres =-seq(br[1]-h/2„h,n+2);freq = c(0, fr, 0)
plot(centres, freq, ylab = "fréquence",axes=F, xlab =
deparse(substitute(x)), xlim =
range(centres),ylim=c(0,max(fr)+3))
axis(1,at=centres, pos=0); axis(2,at=range(c(0,fr)))
text(centres,freq+2,freq)
title(main="Polygone de fréquence",cex=1.5)
polygon(centres, freq, density=15,angle=45,col = 2) }
```

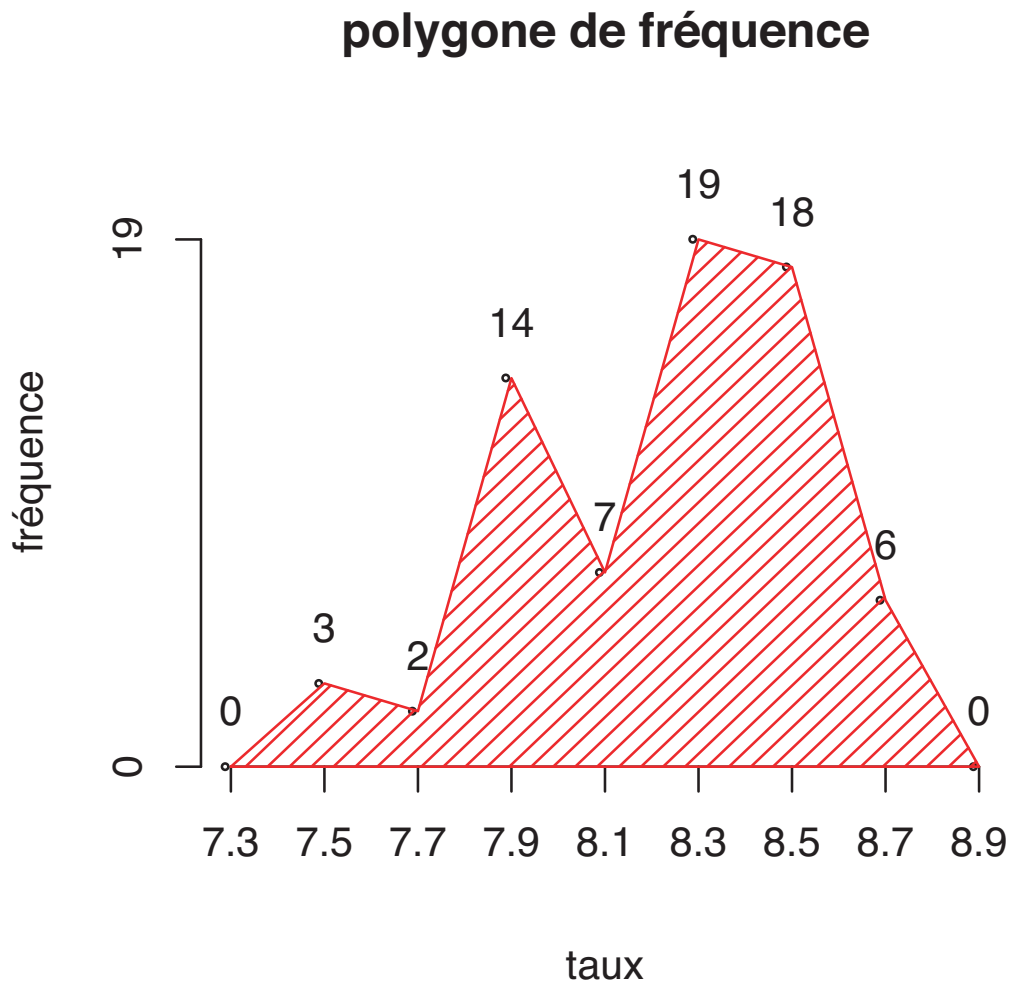
Polygone de fréquence (suite)

- La fonction `freq.poly`

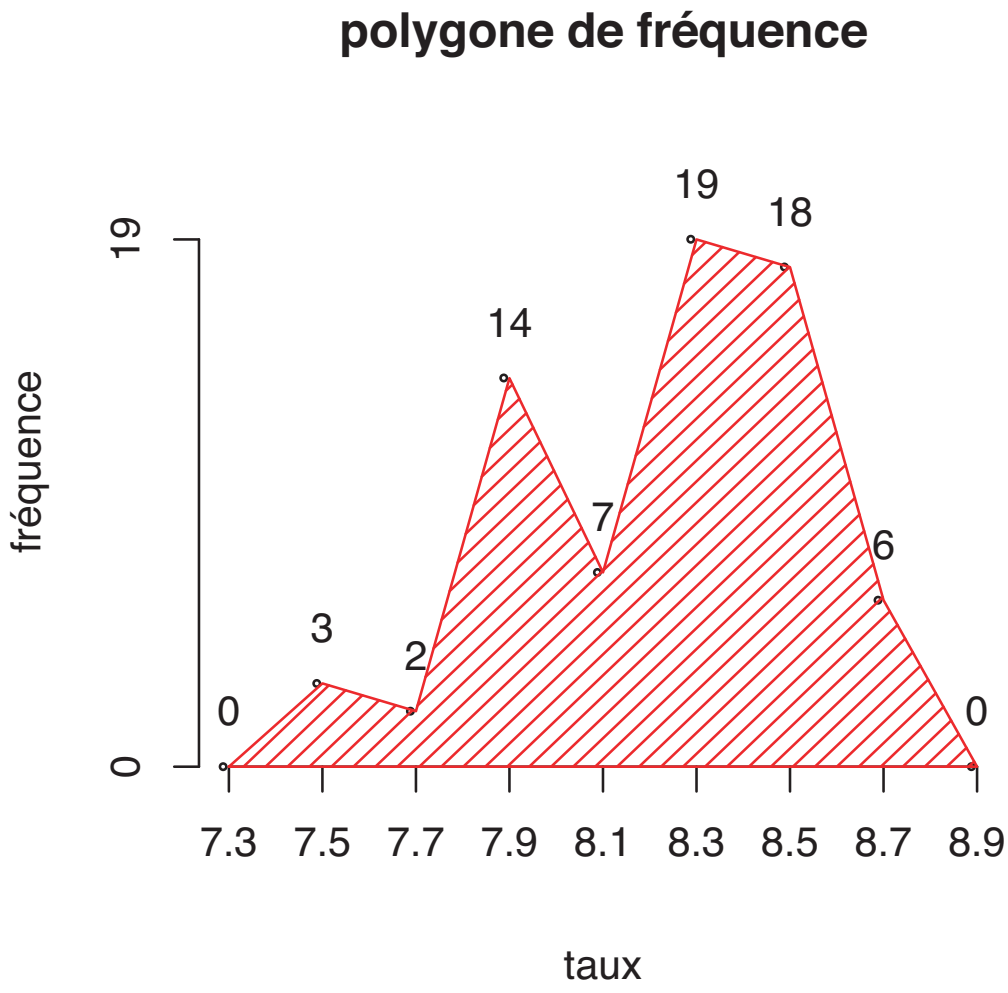
```
freq.poly function(x, br=NULL ) {
if(missing(br)) br = hist(x, plot = F)$breaks
fr = hist(x, , br, plot = F)$counts
h=br[2]-br[1]; n = length(fr)
centres =-seq(br[1]-h/2„h,n+2);freq = c(0, fr, 0)
plot(centres, freq, ylab = "fréquence",axes=F, xlab =
deparse(substitute(x)), xlim =
range(centres),ylim=c(0,max(fr)+3))
axis(1,at=centres, pos=0); axis(2,at=range(c(0,fr)))
text(centres,freq+2,freq)
title(main="Polygone de fréquence",cex=1.5)
polygon(centres, freq, density=15,angle=45,col = 2) }
```

- > freq.poly(taux)

Polygone de fréquence (suite)



Polygone de fréquence (suite)



- Au-dessus des points milieux des intervalles, on a inscrit les fréquences correspondantes.