Esercitazione 1

September 30, 2024

1 Esercitazione 1

1.1 Qualche risultato che vi serve per gli esercizi

Il Monte Carlo vale anche per funzioni di più variabile:

$$E(h(X,Y)) = \int_{\mathcal{X}} \int_{\mathcal{Y}} h(x,y) f(x,y) d\lambda(y) d\lambda(x) \approx \frac{\sum_{i=1}^{n} h(x_i, y_i)}{n}$$

e anche nel caso in cui $h(x,y) = h^*(x)$:

$$E(h^*(X)) = \int_{\mathcal{X}} h^*(x) f(x) d\lambda(x) = \int_{\mathcal{X}} \int_{\mathcal{Y}} h(x,y) f(x,y) d\lambda(y) d\lambda(x) \approx \frac{\sum_{i=1}^{n} h(x_i, y_i)}{n} = \frac{\sum_{i=1}^{n} h^*(x_i)}{n}$$

Potete simulare da una congiunta

$$f(x^1, x^2, \dots, x^p) = f(x^1) f(x^2|x^1) f(x^3|x^2, x^1) \dots f(x^p|x^{p-1}, x^{p-2}, \dots, x^2, x^1)$$

1. simulando x^1 da $f(x^1)$,
2. poi x^2 da $f(x^2|x^1)$,
3. poi x^3 da $f(x^3|x^2, x^1)$
4. e poi ….,
5. e x^p da $f(x^p|x^{p-1}, x^{p-2}, \dots, x^2, x^1)$

1.2 Stime di densità

Dato un campione $(x_1, \dots x_n)$ potete ottenere una stima della densità sottostante in un punto a come

$$\hat{f}(a) = \frac{\sum_{i=1}^{n} k(x_i - a)}{n} \approx \int_{\mathcal{X}} k(x-a) f(x) d\lambda(x)$$

dove $k(x,a)$ è un kernel, che generalmente ha le seguenti proprietà

1. $k(x-a) > 0$ per tutti
2. simmetrico rispetto a zero

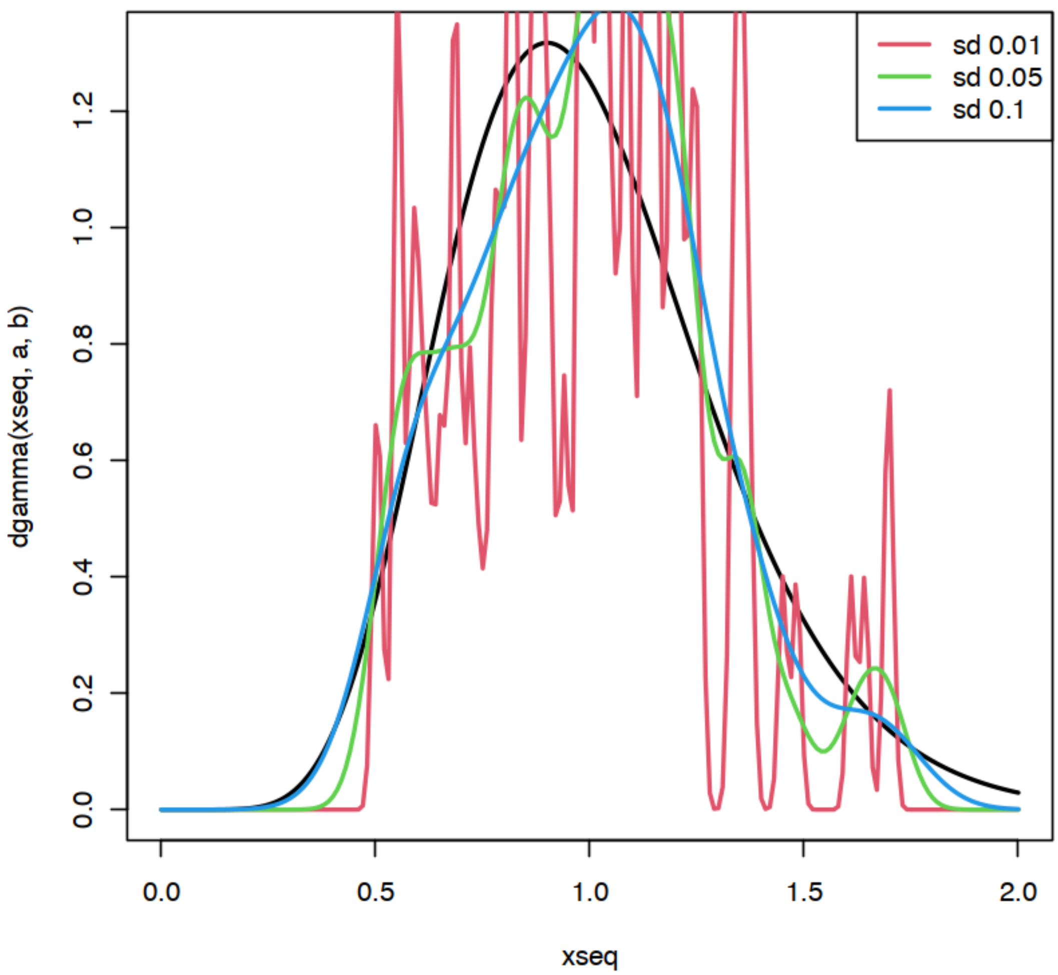
Questo è quello che fa il comando density() di R. L'integrale è di fatto una convoluzione tra $f(x)$ e il kernel \$ k(x - a)\$.

Adesso assumete che $X \sim G(10, 10)$, simulate $n = 10$ osservazioni e stimate con il metodo del kernel la densità sottostante assumendo un kernel gaussiano

$$k(x_i - a) = (2\pi\sigma^2)^{-0.5} \exp(-\frac{1}{2\sigma^2}(x_i - a)^2)$$

(il kernel è la densità di una normale con media a, valutata in x_i) vedete le differenze se cambiate il valore di $\sigma \in \{0.01, 0.05, 0.1\}$ e confrontatelo graficamento con la vera densità

[1]:
```
# simulazione
n = 100
a = 10
b = 10
x = rgamma(n,a,b)

# scegliamo i parametri di un kernel gaussiano
# con diversi parametri
stand_dev_1 = 0.01
stand_dev_2 = 0.05
stand_dev_3 = 0.1

# calcoliamo la stima di densità in xseq
xseq = seq(0,2, by=0.01)

stima_dens_1 = c()
stima_dens_2 = c()
stima_dens_3 = c()
for(i in 1:length(xseq))
{
  stima_dens_1[i] = sum( dnorm(x,xseq[i],stand_dev_1)  )/n
  stima_dens_2[i] = sum( dnorm(x,xseq[i],stand_dev_2)  )/n
  stima_dens_3[i] = sum( dnorm(x,xseq[i],stand_dev_3)  )/n
}

par(mfrow=c(1,1))
plot(xseq, dgamma(xseq,a,b), type="l", lwd=2)
lines(xseq,stima_dens_1, col=2, lwd=2)
lines(xseq,stima_dens_2, col=3, lwd=2)
lines(xseq,stima_dens_3, col=4, lwd=2)
legend("topright", paste("sd ", c(stand_dev_1,stand_dev_2,stand_dev_3),␣
 ↪sep=""), col=2:4, lwd=2, lty=1 , cex=1)
par(mfrow=c(1,1))
```



1.3 Teorema del limite centrale

Assumete che $X \sim P(\lambda)$ e definite

$$\bar{X}_n = \sum_{i=1}^{n} X_i/n$$

Verificate se la distribuzione di

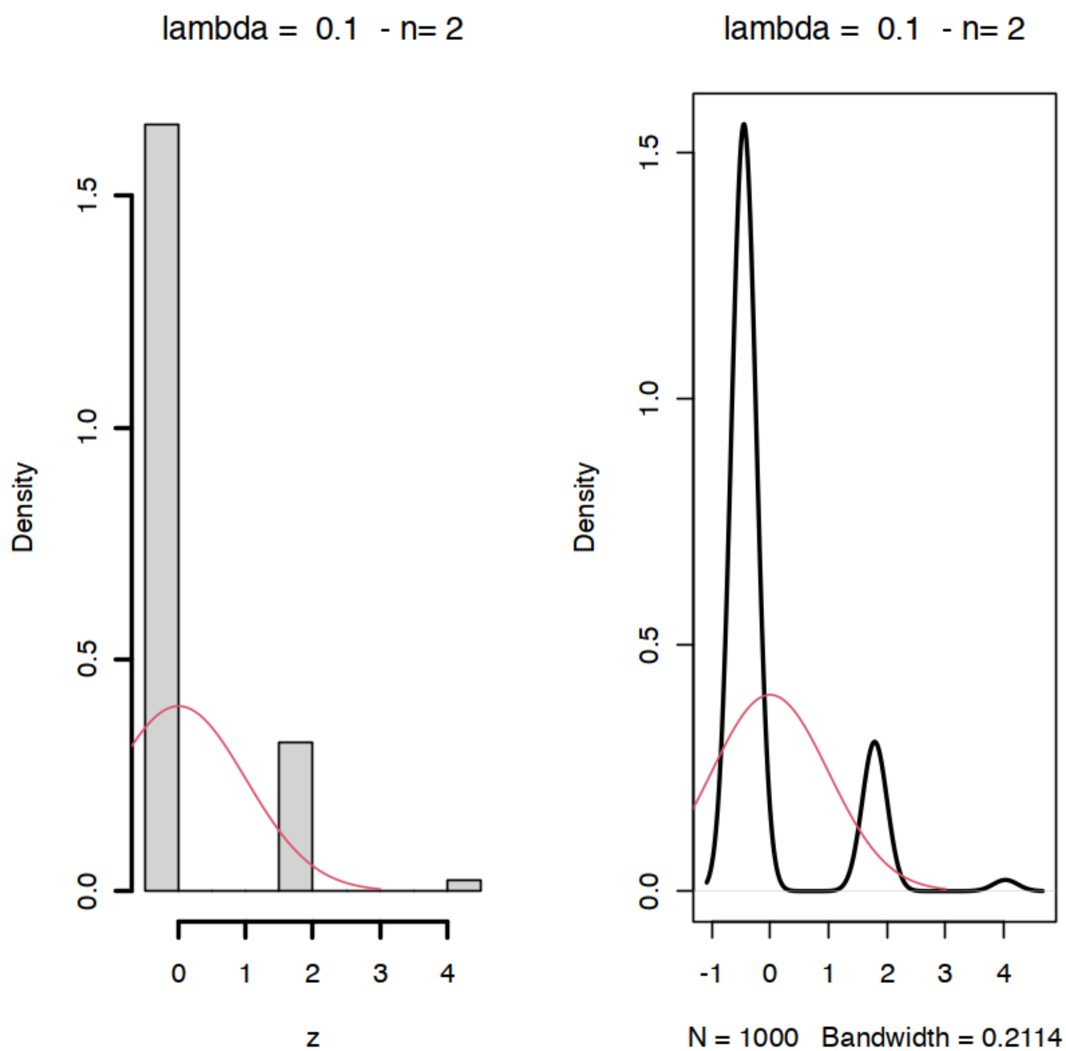
$$Z = \frac{\bar{X}_n - E(X)}{\sqrt{\frac{Var(X)}{n}}}$$

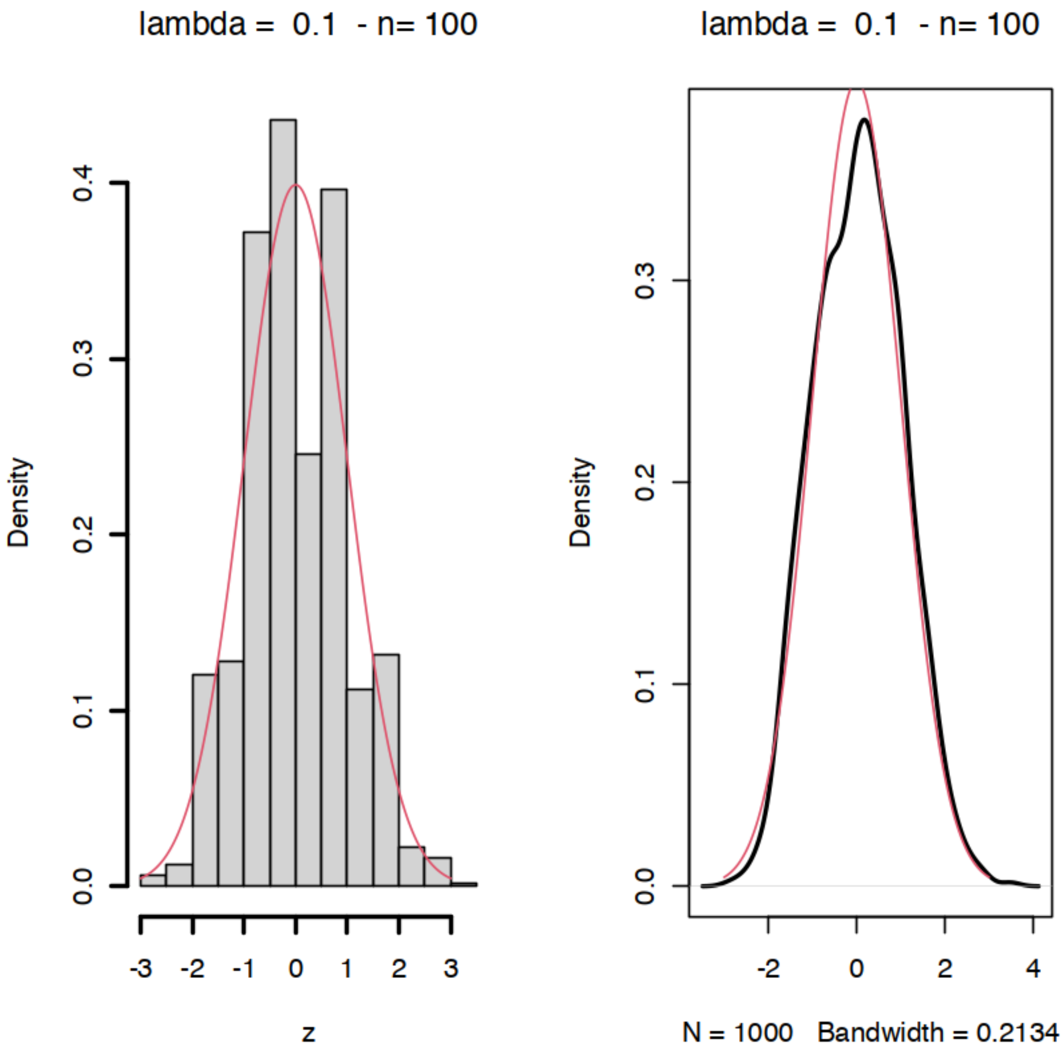
segue approssimativamente un $N(0, 1)$ con $n = 2, 100, 10000$.

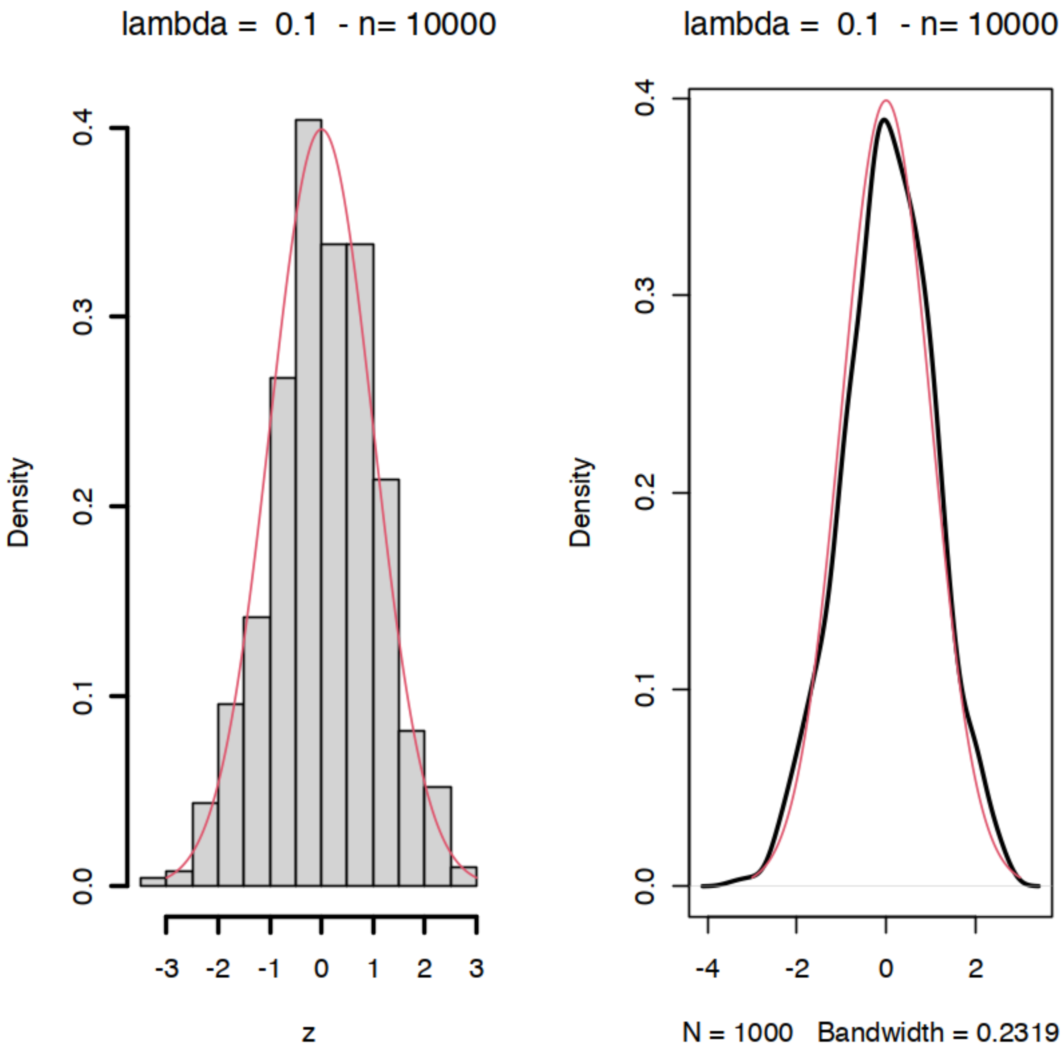
Provate con $\lambda = 0.1$ e $\lambda = 10$, e plottate le distribuzione di Z usando un istogramma e il comando density()

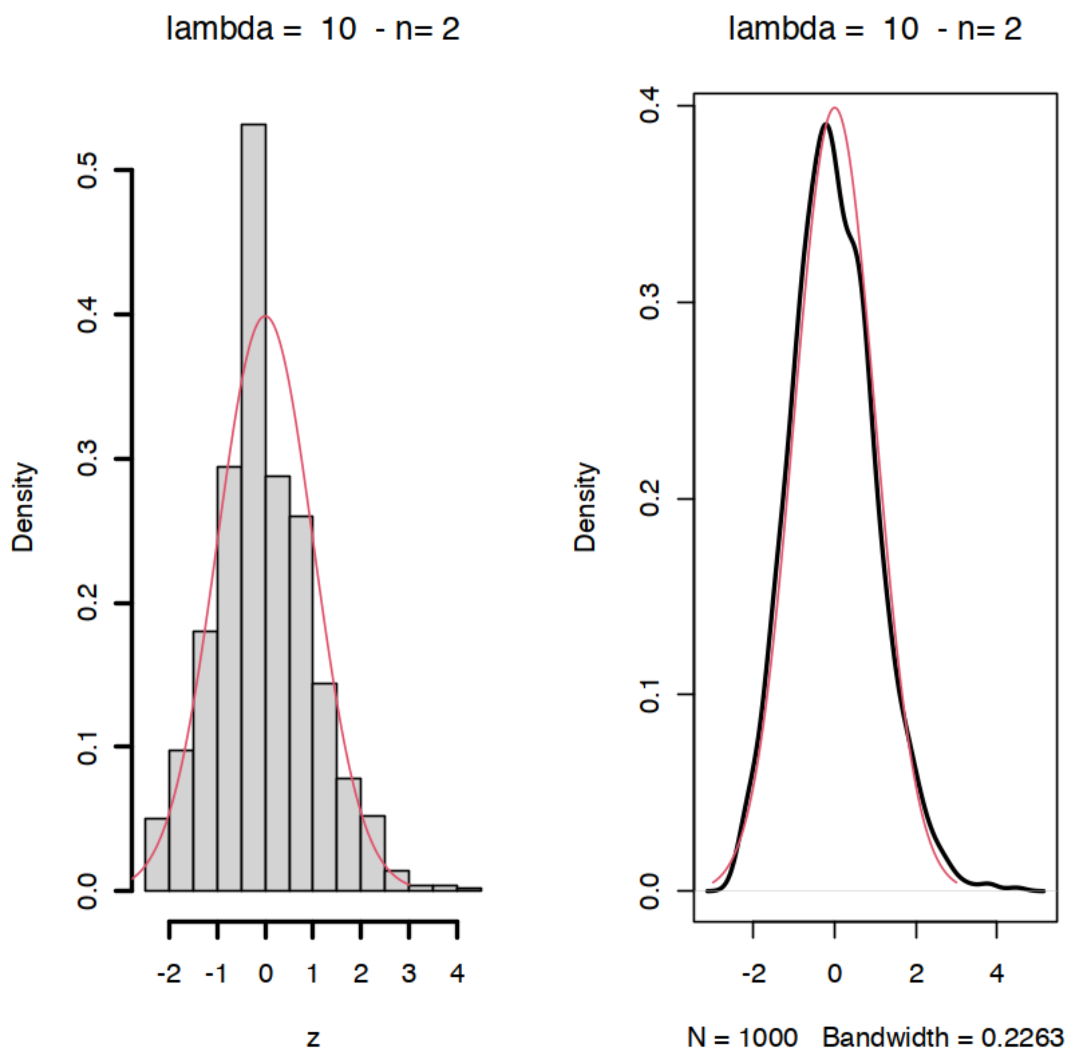
[2]:

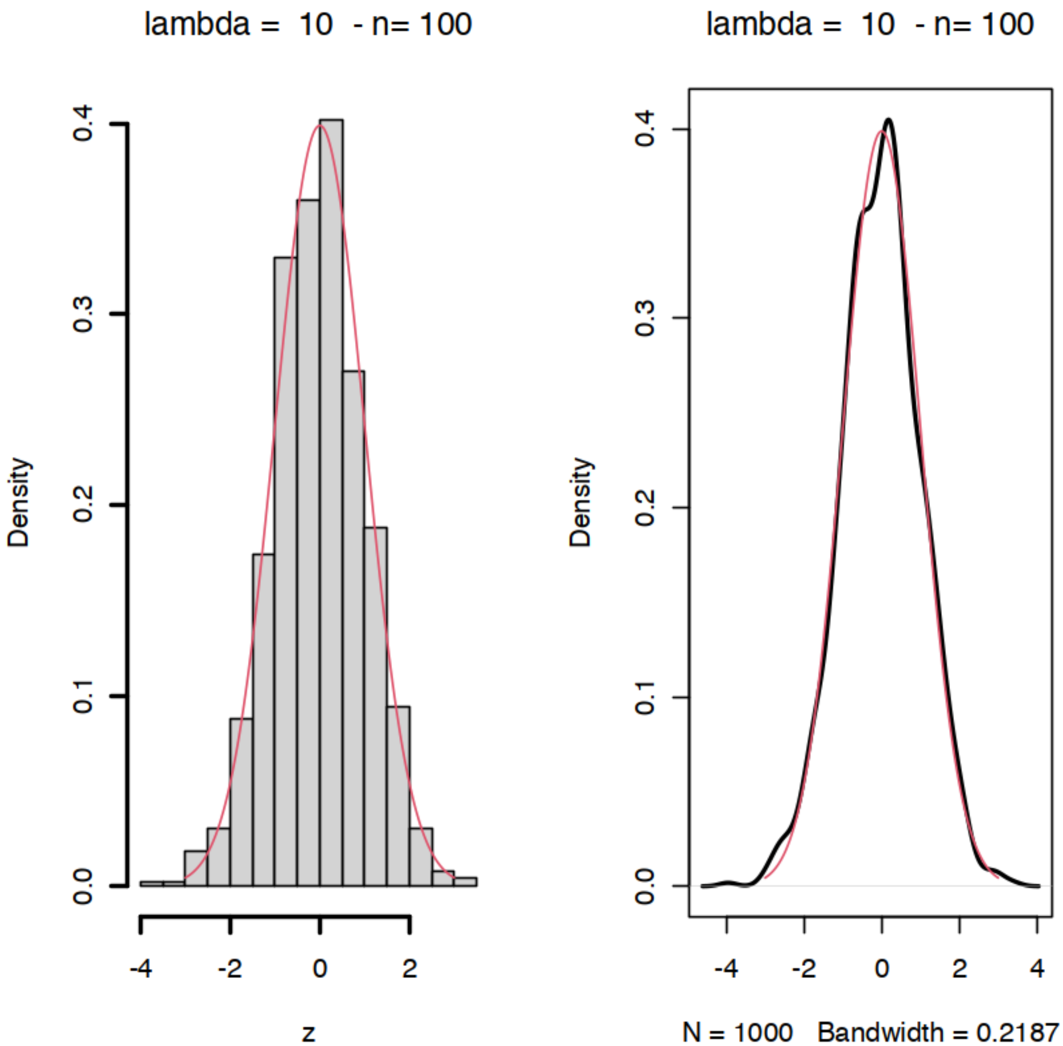
```
nsim = 1000
n_gen = 10000


for(lambda in c(0.1,10))
{
    x_gen = matrix(rpois(n_gen*nsim, lambda), ncol=nsim)
    norm_val = seq(-3,3,by = 0.01)
    par(mfrow=c(1,2))
    for(select_n in 1:3)
    {
        n = c(2,100,10000)[select_n]
        x = x_gen[1:n,]
        xn = colMeans(x)
        z = (xn - lambda)/sqrt(lambda/n)

        hist(z, freq = F, main= paste("lambda = ", lambda ," - n=", n), lwd = 2)
        lines(norm_val, dnorm(norm_val, 0, 1), col=2)
        plot(density(z ), main= paste("lambda = ", lambda ," - n=", n),, lwd =
 ↪2)
        lines(norm_val, dnorm(norm_val, 0, 1), col=2)


    }
    par(mfrow=c(1,1))
}
```


lambda = 0.1 - n= 2
Density
0.0
0.5
1.0
1.5
0
1
2
3
4
z
lambda = 0.1 - n= 2
Density
0.0
0.5
1.0
1.5
-1
0
1
2
3
4
N = 1000 Bandwidth = 0.2114

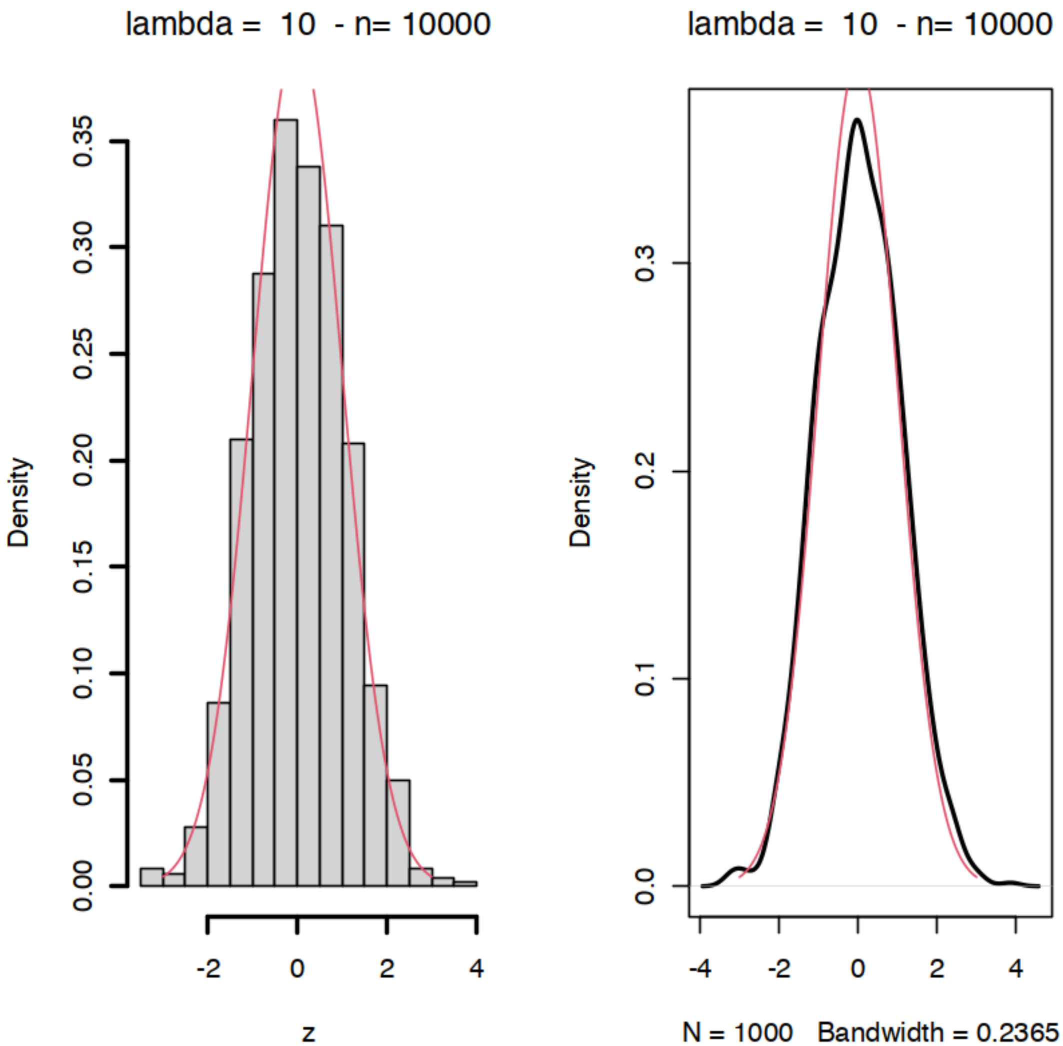

lambda = 0.1 - n= 100
Density
z
lambda = 0.1 - n= 100
Density
N = 1000 Bandwidth = 0.2134


lambda = 0.1 - n= 10000
Density
0.0
0.1
0.2
0.3
0.4
-3
-2
-1
0
1
2
3
z
lambda = 0.1 - n= 10000
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
N = 1000 Bandwidth = 0.2319


lambda = 10 - n= 2
Density
0.0
0.1
0.2
0.3
0.4
0.5
-2
-1
0
1
2
3
4
z
lambda = 10 - n= 2
Density
0.0
0.1
0.2
0.3
0.4
-2
0
2
4
N = 1000 Bandwidth = 0.2263



lambda = 10 - n= 100
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
z
lambda = 10 - n= 100
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
N = 1000 Bandwidth = 0.2187



1.4 Beta binomiale

Assumete che

$$X|p \sim Bin(m,p) \qquad p \sim Beta(a,b)$$

con m intero, $p \in (0,1)$ e $X \in \{0,1,\dots m\}$.

è noto che la marginale di X è una beta-binomiale di parametri (m,a,b)

$$X \sim BetaBin(m,a,b)$$

che ha pmf

$$f(x) = P(X = x) = \binom{m}{x} \frac{B(a+x, b+n-x)}{B(a,b)}$$

dove il coefficiente binomiale è

$$\binom{m}{x} = \frac{m!}{x!(m-x)!}$$

Inoltre sappiamo che

$$E(X) = m\frac{a}{a+b}$$

mentre

$$E(X|p) = mp$$

1. Stimare i valori di $f(x)$ nei punti $\{0, 1, \dots, m\}$ usando la distribuzione condizionate di $X|p$.
2. Scrivete una funzione che calcoli la marginale e verificate graficamente che la f() vera e quella stimata siano simili
3. stimare il valore atteso di X sia usando

   $$E(X) = \int xf(x)d\lambda(x)$$

   sia

   $$E_p(E_x(X|p)) = E(X) = \int E_x(X|p)f(p)d\lambda(p)$$

   e verificare quale dei due è più variabile usando $n = 10$ e $n = 1000$.

Per la simulazione di una beta-binomiale usate il pacchetto TailRank, che va installato usando install.packages("TailRank"). Se il pacchetto Biobase vi da problemi, usate i comandi

install.packages("BiocManager")

BiocManager::install("Biobase")

Alternativamente, potete vedere nell'esercizio "Congiunte e marginali" che se simulate p_i dalla sua marginale $B(a, b)$, e poi x_i da $Binom(m, p)$, allora x_i è un **campione dalla marginale BetaBinomial**

Stima varianza

Per la stima della varianza di

$$\hat{E}(h(X)) = \frac{\sum_{i=1}^{n} h(x_i)}{n}$$

avete due possibilità.

Come prima possibilità potete scrivere

$$Var(\hat{E}(h(X))) = \frac{\sum_{i=1}^{n} Var(h(x_i))}{n^2} = \frac{Var(h(x))}{n}$$

e stimare $Var(h(x))$ con

$$Var(h(x)) \approx \frac{\sum_{i=1}^{n^*} (h(x_i^*) - E(h(x^*)))^2}{n^*}$$

dove ho utilizzato lo $*$ per indicare che potete usare o no i gli stessi campioni usati per calcolare $\hat{E}(h(X))$ oppure prenderne dei nuovi. Inoltre non conoscete $E(h(x^*))$ e dovete stimarlo

La seconda opzione è di definire

$$W = \hat{E}(h(X))$$

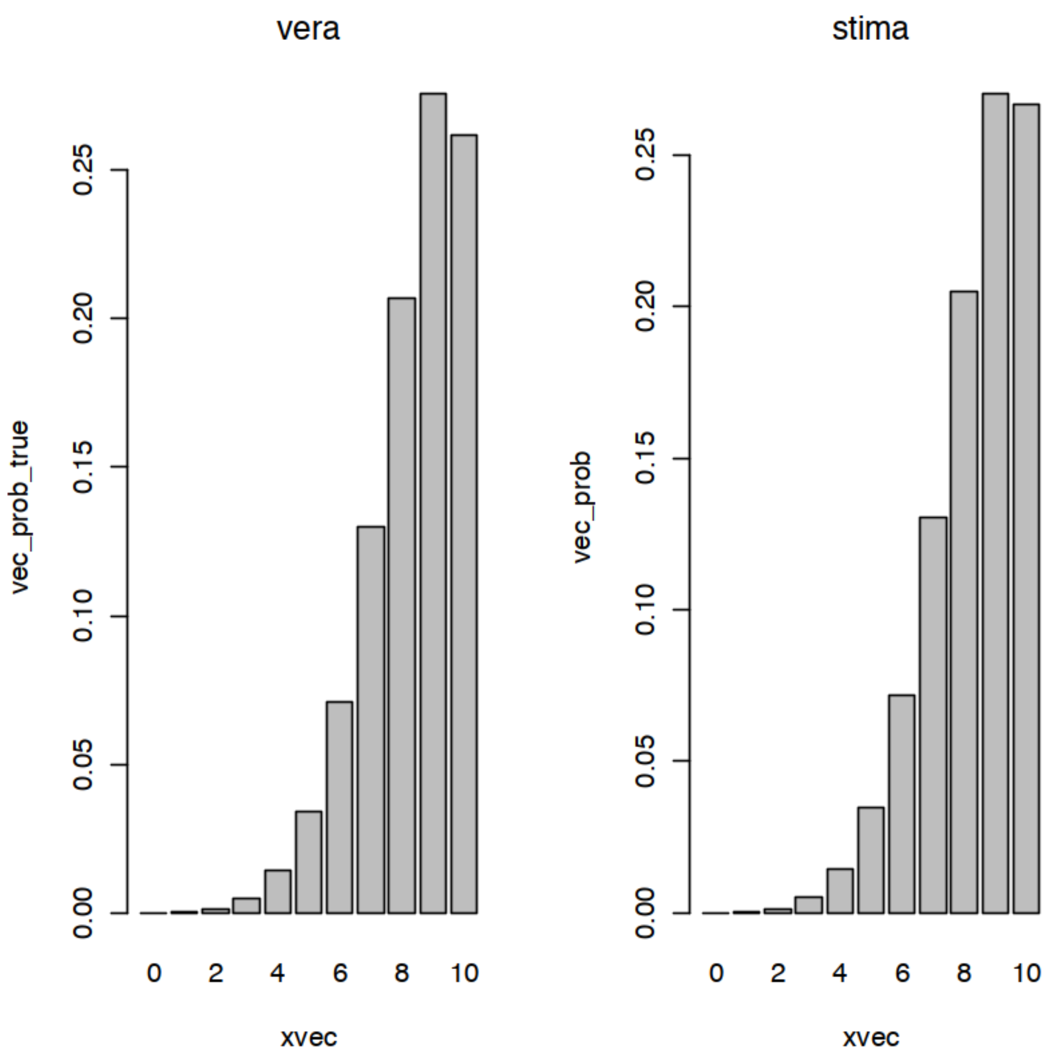
e calcolare

$$Var(W) \approx \frac{\sum_{i=1}^{n^{**}} (W_i^{**} - E(W))^2}{n^{**}}$$

quindi usando campioni di W invece che di X. Anche qui dovete stimare $E(W)$

[3]:
```
# densità della betabinomiale
dbetabin = function(x, m,a,b)
{
    bin_coef = factorial(m)/(factorial(x)*factorial(m-x))
    return(  bin_coef * beta(a+x, b+m-x)/beta(a,b))
}
# parametri
a =10
b = 2
m = 10
n = 1000

# dove valuto la probabilità
xvec = 0:10
p = rbeta(n, a,b)
# dove salvo i valori delle prob stimate con il monte carlo
vec_prob = rep(NA, length(xvec))
# dove salvo i valori delle prob vere
vec_prob_true = rep(NA, length(xvec))
for(i in 1:length(xvec))
{
    vec_prob[i] = mean(dbinom(xvec[i], m, prob = p))
    vec_prob_true[i] = dbetabin(xvec[i], m, a, b)
}
```

[4]:
```
par(mfrow=c(1,2))
barplot(vec_prob_true ~ xvec, main="vera")
barplot(vec_prob ~ xvec, main="stima")
par(mfrow=c(1,1))
```

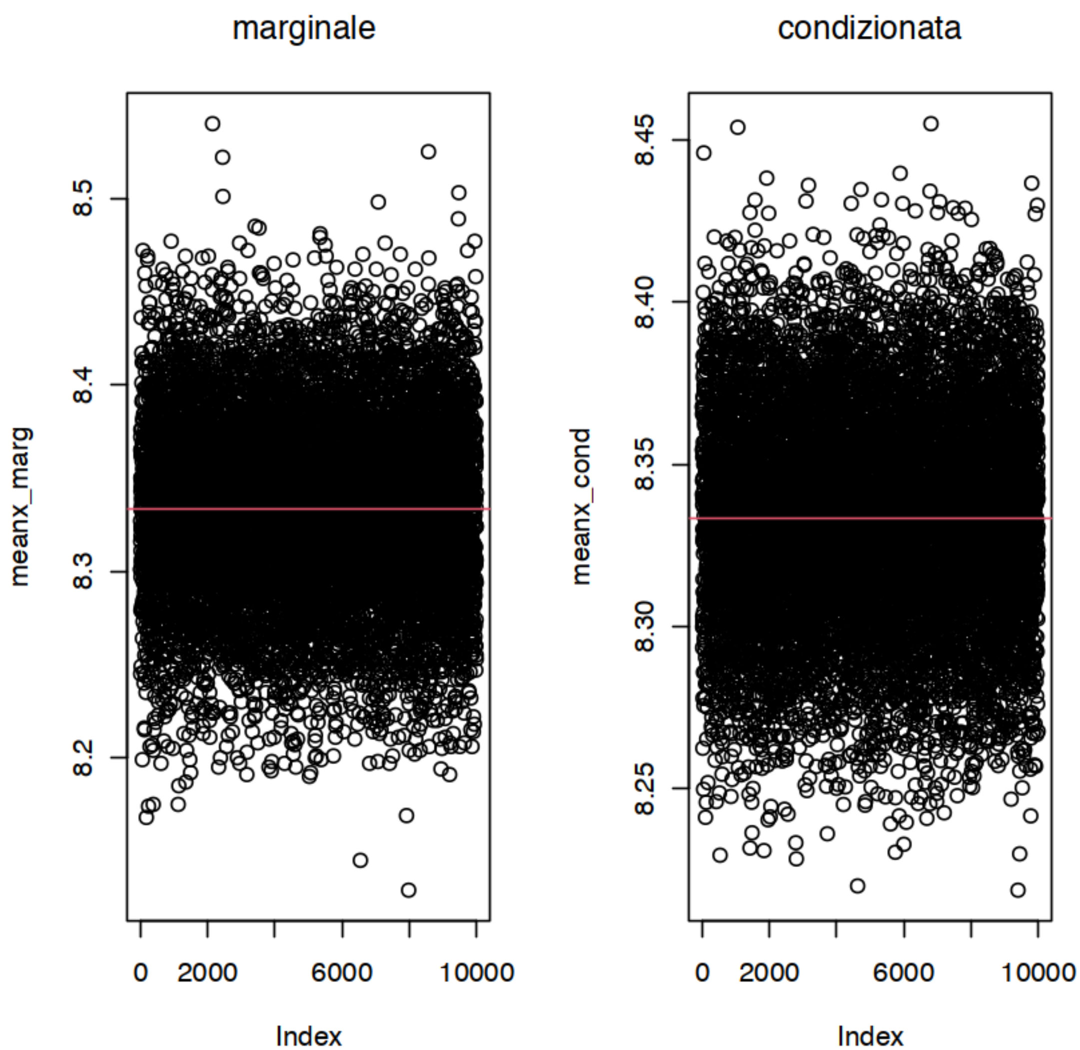


[5]:
```
## punto 3
library(TailRank)

nsim = 10000
n = 1000
meanx_marg = rep(NA, nsim)
meanx_cond = rep(NA, nsim)

for(isim in 1:nsim)
{
    meanx_marg[isim] = mean(rbb(n,m,a,b))

    p = rbeta(n, a,b)
```

```
    meanx_cond[isim] = mean(m*p)
}
```

```
Caricamento del pacchetto richiesto: oompaBase
```

[6]:
```
par(mfrow=c(1,2))

plot(meanx_marg, main=paste("marginale " ))
abline(h =m*a/(a+b), col=2)
plot(meanx_cond, main=paste("condizionata"))
abline(h =m*a/(a+b), col=2)
par(mfrow=c(1,1))
```



[7]:
```r
### calcoliamo le varianze della marginale

nstar = 1000
nstarstar = 1000
## metodo 1 - marginale
xstar = rbb(n, m, a, b)
ex_marg = mean(xstar)
var_marg = sum((xstar-mean(xstar))^2)/n
stimatore1_marg = var_marg/n

## metodo 2 - marginale
vec_mean_marg = rep(NA, nstarstar)
for(isim in 1:nstarstar)
{
    xstar = rbb(n,m,a,b)
    vec_mean_marg[isim] = mean(xstar)
}
stimatore2_marg = sum( (vec_mean_marg- mean(vec_mean_marg))^2 )/n
stimatore1_marg
stimatore2_marg


## metodo 1 - condizionata
p = rbeta(n, a,b)
cond_mean = m*p # h(x)
ex_cond = mean(cond_mean)
var_cond = sum((cond_mean-mean(cond_mean))^2)/n
stimatore1_cond = var_cond/n

## metodo 2 - condizionata
vec_mean_cond = rep(NA, nstarstar)
for(isim in 1:nstarstar)
{
    p = rbeta(n, a,b)
    cond_mean = m*p # h(x)
    vec_mean_cond[isim] = mean(cond_mean)
}
stimatore2_cond = sum( (vec_mean_cond- mean(vec_mean_cond))^2 )/n
stimatore1_cond
stimatore2_cond
```

0.002285959

0.002445013504

0.00104290037163057

0.00105210570785163

1.5 Congiunte e marginali

Come nel punto precedente, assumete che

$$X|p \sim Bin(m,p) \qquad p \sim Beta(a,b)$$

Simulate n campioni dalla congiunta

$$f(x,p) = f(x|p)f(p)$$

simulando $p_i \sim Beta(a,b)$ e poi $X_i|p_i \sim Bin(m,p_i)$

Verificate che i campioni $(x_1,\dots,x_n)$ provengono dalla marginale di X (questo vi da anche un modo facile per simualre da una marginale quando si conoscono le opportune condizionate). Potete usare il seguente risultato

$$P(X \in A) = \int_A f(x)\lambda(x) = \int_{\mathcal{X}} 1_A(x)f(x)d\lambda(x)$$

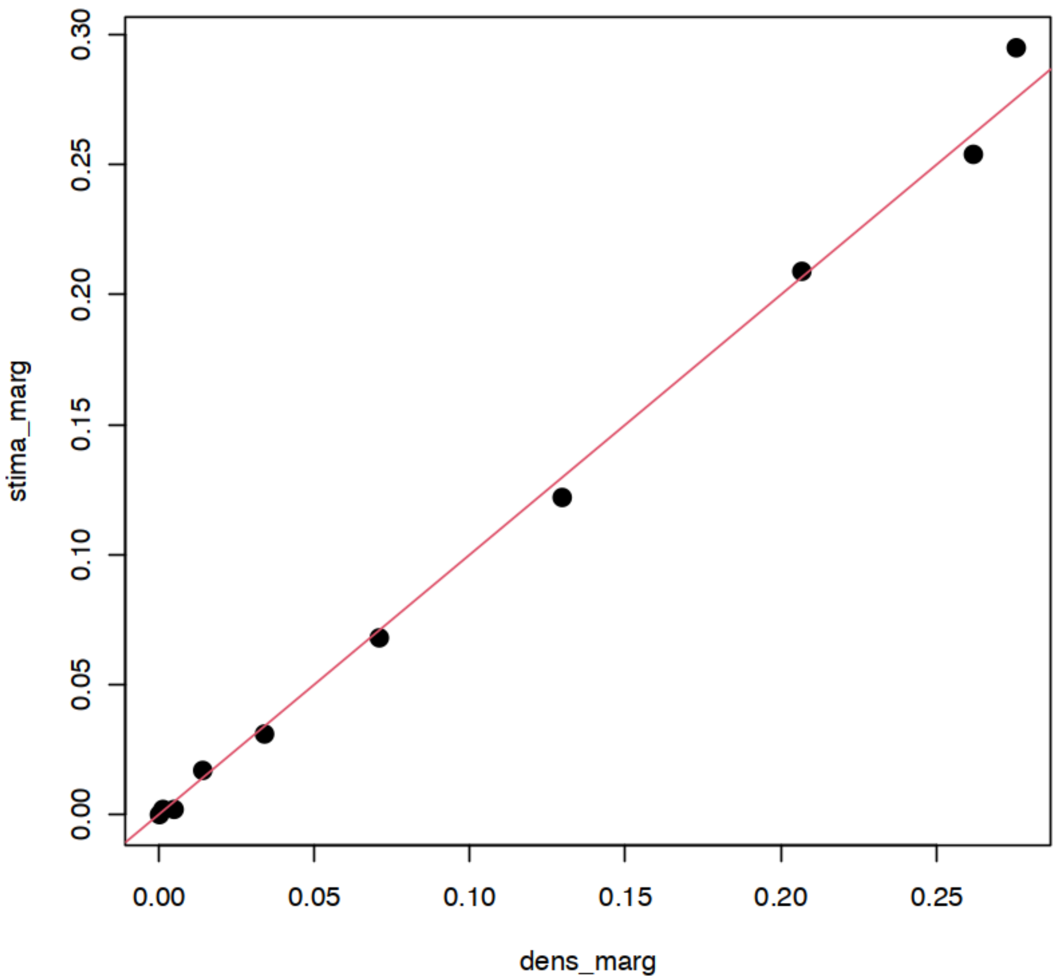
per stimare la $P(X = a)$

Per vedere perchè questa cosa è vera, potete usare

$$P(X \in A) = \int_{\mathcal{X}} 1_A(x)f(x)d\lambda(x) = \int_{\mathcal{X}}\int_p 1_A(x)f(x,p)d\lambda(p)d\lambda(x)$$

[8]:

```
# dove valuto la probabilità
p = rbeta(n, a,b)
x = rep(NA,n)
for(i in 1:n)
{
    x[i] = rbinom(1,m,p[i])
}
stima_marg = c(NA, m+1)
dens_marg = c(NA, m+1)
for(i in 0:m)
{
    stima_marg[i] = mean(x == i)
    dens_marg[i] = dbb(i, m,a,b)
}
plot(dens_marg,stima_marg, pch=20, cex=2)
abline(a=0,b=1, col=2)
```



1.6 Area del cerchio

Sappiamo che un cerchio di raggio unitario nel piano (x, y) ha area

$$2\int_{-1}^{1}\sqrt{1-x^2}dx = 4\int_{0}^{1}\sqrt{1-x^2}dx = \pi$$

Usando Monte Carlo, stimare il valore di π assumendo

1. $x \sim U(-1, 1)$
2. $x \sim B(a, b)$
3. $x \sim U(-1, 1)$ e $y \sim U(0, 1)$

Per risolvere i punti 1 e 2, notate che potete scrivere gli integrali come

$$\int_X g(x)dx = \int_X \frac{g(x)}{f(x)} f(x)dx$$

dove $f(x)$ è una densità che assume valori diversi da zero in corrispondenza dei punti in cui $g(x)$ è diversa da zero, tranne che su un insieme di misura nulla.

Per il punto 3 invece ricordatevi che il Monte Carlo vale anche per funzioni di più variabile

[9]:
```
## punto 1: int 2 sqrt(1-x^2) f(x)/f(x) dx
## con f(x) uniforme -> h(x) = 4 sqrt(1-x^2)
n = 100000
x = runif(n,-1,1)
area_1 =  mean(4*sqrt(1-x^2))
area_1

## punto 2: int 4 sqrt(1-x^2) f(x)/f(x) dx
## con f(x)\sim B(a,b) -> h(x) = 4 sqrt(1-x^2)/f(x)
n = 100000
a = 2
b = 3
x = rbeta(n,a,b)
area_2 =  mean(4*sqrt(1-x^2)/dbeta(x,a,b)   )
area_2

## punto 3: int_{-1}^1 2int_{0}^{sqrt{1-x^2}}  dx = int_{-1}^1 int_{0}^{1} 2 *␣
↪1_{0,sqrt{1-x^2}}(y) f(x,y)/f(x,y)
## con f(x,y) = 1/2 -> h(x) = 4 1_{0,sqrt{1-x^2}}(y)
x = runif(n,-1,1)
y = runif(n,0,1)
area_3 = 4* sum(y < sqrt(1-x^2))/n
area_3
```

3.14435235606541

3.12477244213394

3.13868