LI Yanting

Exercice 1-6 (a)

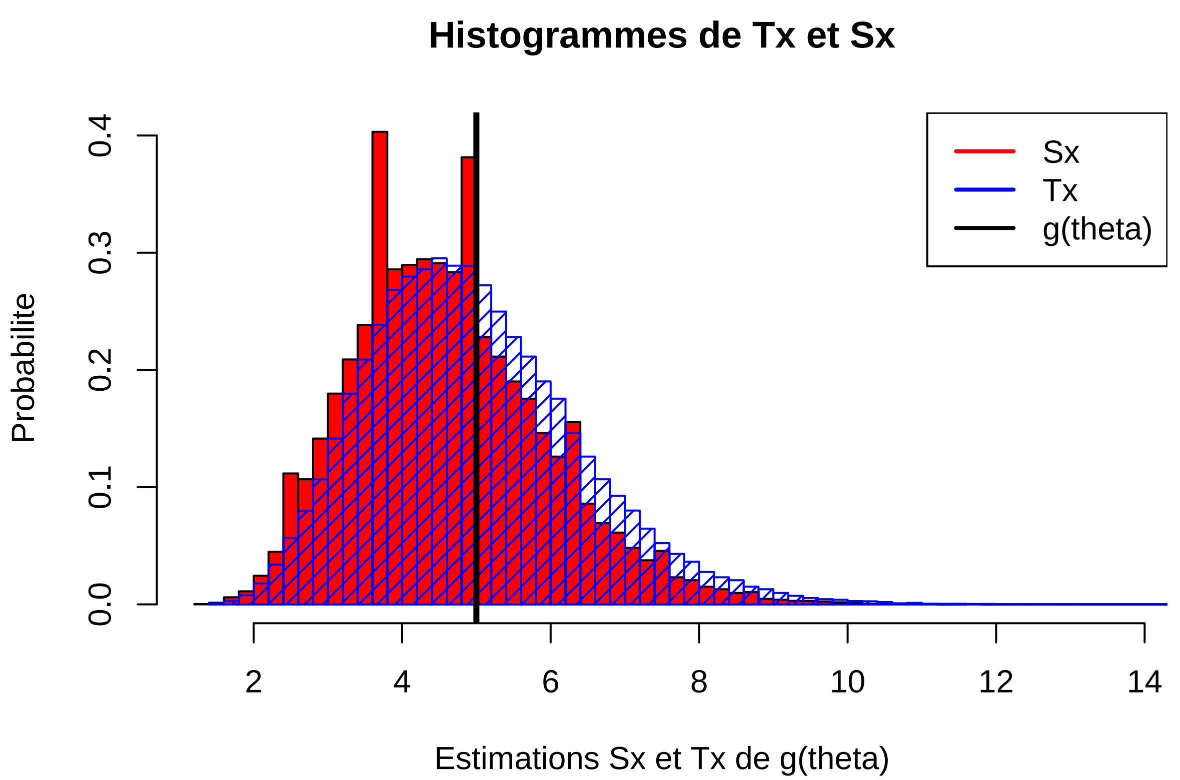
```
set.seed(42,kind="Marsaglia-Multicarry");
#Ex.1-6 (a)
M = 1000000;
v=rgeom(M,prob=0.2)+1;
Z=matrix(v,nrow = 10);
theta_0=0.2;
```

(b)

```
h = 10/10.8
Tx <- apply(Z,2,mean)
#apply(Z,2,mean):  2--colonne
Sx <- h*Tx
```

(c)

```
hist(Sx, col="red",probability=TRUE,
     main="Histogrammes de Tx et Sx",
     breaks=50,
     xlab= "Estimations Sx et Tx de g(theta)",
     ylab ="Probabilite")
hist(Tx,col="blue", probability=TRUE, add=TRUE, density=15, breaks=50)
abline(v = 1/theta_0, lwd=3)
legend("topright", lwd=2, col=c("red", "blue", "black"),
       legend=c( "Sx", "Tx", "g(theta)"))
```



```r
mean(Sx)
```

```
## [1] 4.627897
```

```r
mean(Tx)
```

```
## [1] 4.998129
```

(d)

```r
#g_theta = 1/theta_0

cal_L1 = function(c){
  r=(mean(c)-1/theta_0)^2
  return(r)
}
cal_Lh = function(c){
  r=(h*mean(c)-1/theta_0)^2
  return(r)
}
L1=apply(Z,2,cal_L1)
Lh=apply(Z,2,cal_Lh)
```

(e)

```r
t = 0.2
n = 10
R_Sh = 1/t^2 - 2*h/t * 1/t + h^2*((1-t)/n*t^2+1/t^2)
R_t = 1/t^2 - 2/t^2 + (1-t)/(n*t^2) + 1/t^2
R_T = mean(L1)
R_S = mean(Lh)

R_t
```

```
## [1] 2
```

```r
R_Sh
```

```
## [1] 0.1399177
```

```r
R_T
```

```
## [1] 1.992096
```

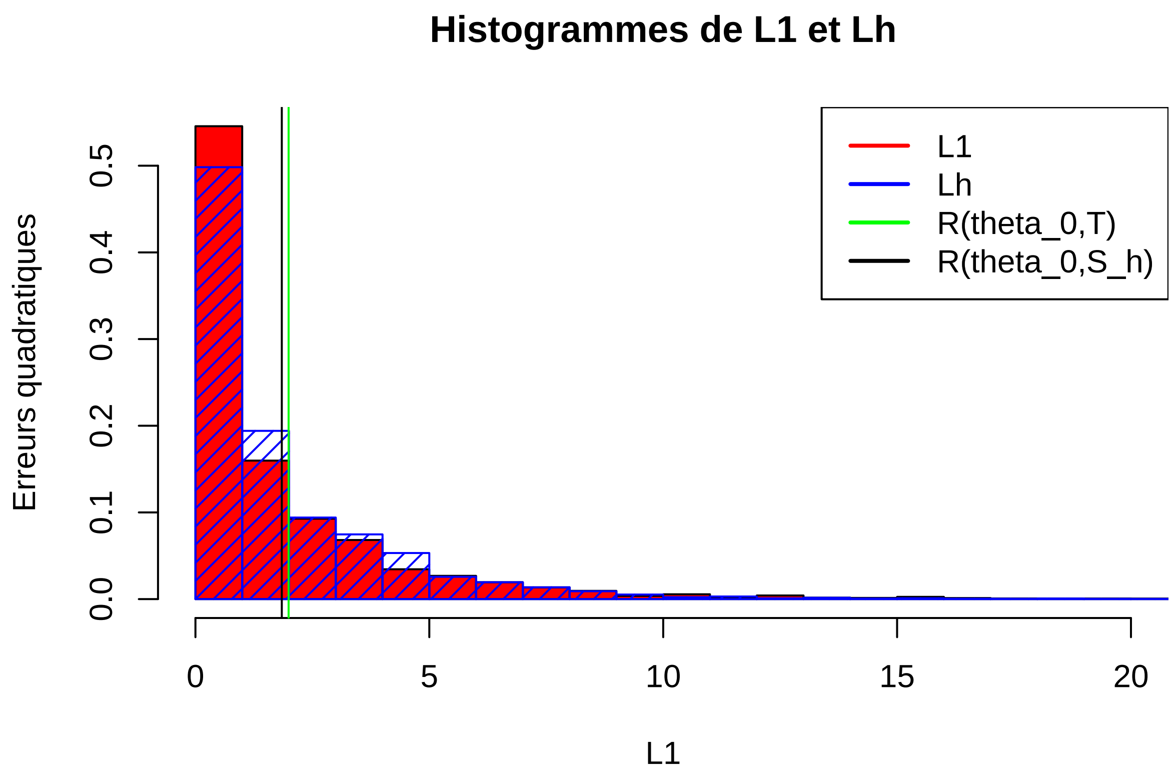
```r
R_S
```

```
## [1] 1.846359
```

(f)

```r
hist(L1, col="red",
     probability=TRUE,
     main="Histogrammes de L1 et Lh",
     breaks=75,
     ylab ="Erreurs quadratiques",
     xlim=range(0,20))

hist(Lh,col="blue", probability=TRUE,
     add=TRUE, density=15, breaks=75)
abline(v = R_T, lwd=1,col='green')
abline(v = R_S, lwd=1,col='black')
legend("topright", lwd=2, col=c("red", "blue", "green", "black"),
       legend=c( "L1", "Lh", "R(theta_0,T)","R(theta_0,S_h)"))
```



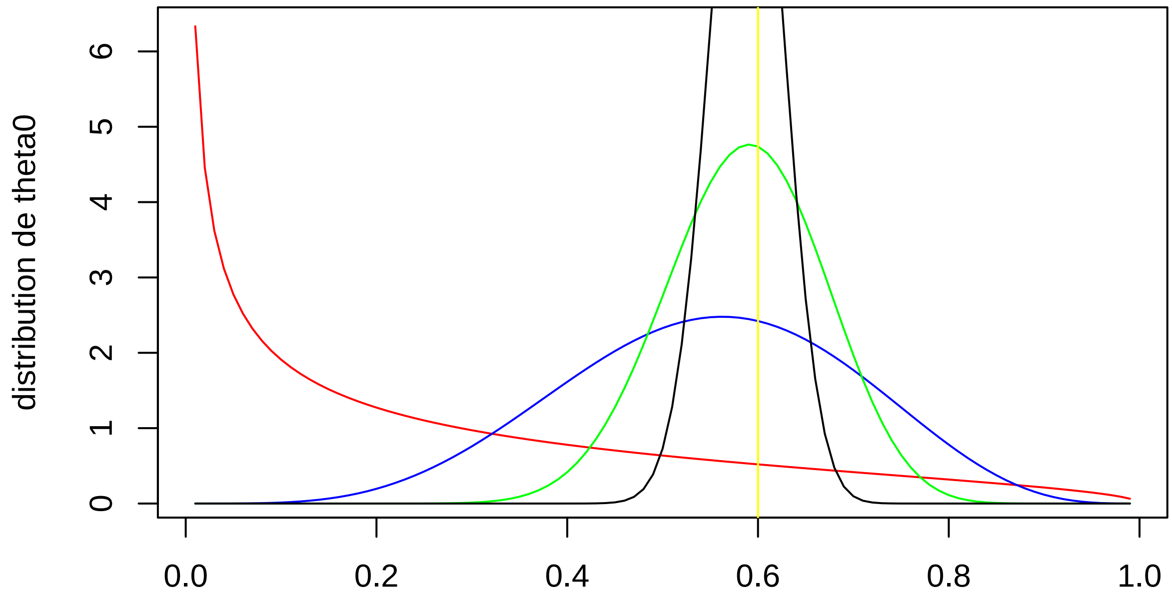
Exercice 2-4 (a)

```r
grille = seq(0, 1, by = 0.01)
L = length(grille)
grille = grille[-c(1,L)]
```

(b)

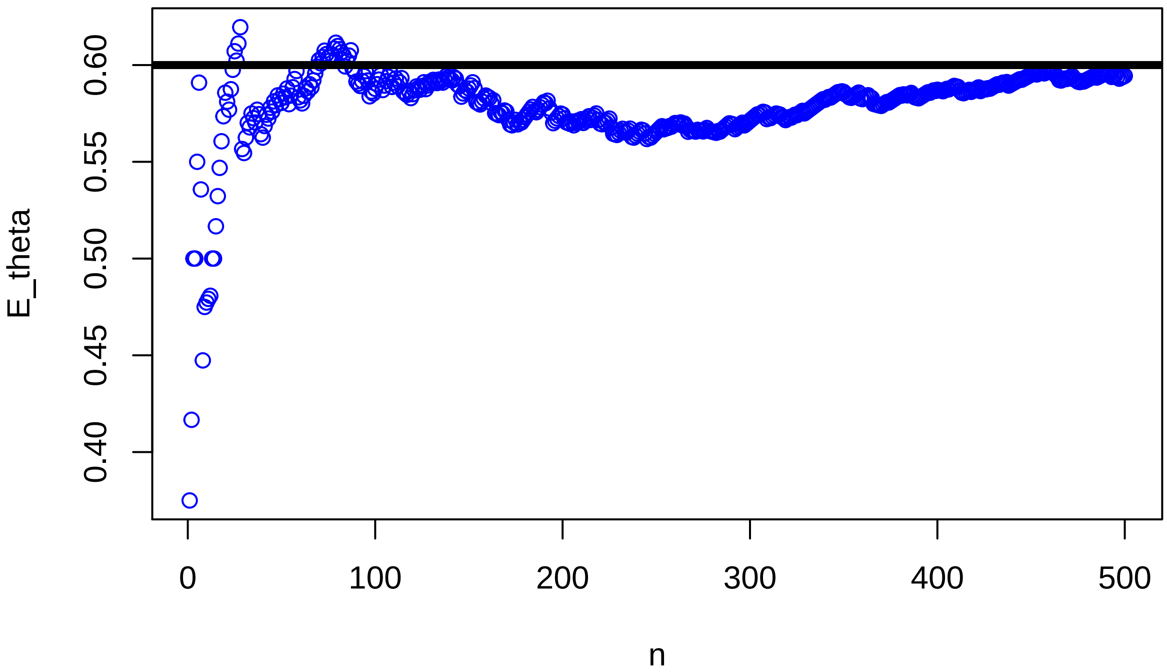
```r
X = rgeom(500,prob=0.6)+1
prior=dbeta(grille,0.5,1.5)
plot(grille,prior,type ='l',col='red',xlab=" ",ylab="distribution de theta0")
n = 5
X0 = dbeta(grille,0.5+n, sum(X[1:n])+1.5 - n)
n = 20
```

```r
X1 = dbeta(grille,0.5+n, sum(X[1:n])+1.5 - n)
n = 100
X2 = dbeta(grille,0.5+n, sum(X[1:n])+1.5 - n)
lines(grille, X0, col = 'blue')
lines(grille, X1, col = 'green')
lines(grille, X2, col = 'black')
abline(v = 0.6, lwd=1, col='yellow')
```



(c)

```r
Cumsumx=cumsum(X)
post=seq(0,0,length.out = 500)
for(i in 1:500){
  post[i]=(0.5 + i)/(0.5 + 1.5 + Cumsumx[i])
}
plot(post,xlab="n",ylab="E_theta",col='blue')
abline(h=0.6,lwd="4")
```



3. 3-3

```
alpha=5/100
A_10=-qnorm(alpha/2)/sqrt(10)
A_10
```

```
## [1] 0.619795
```

```
A_100=-qnorm(alpha/2)/sqrt(100)
A_100
```

```
## [1] 0.1959964
```

```
A_1000=-qnorm(alpha/2)/sqrt(1000)
A_1000
```

```
## [1] 0.0619795
```

3-4

```
mu=0.1
exp(mu+1/2)
```

```
## [1] 1.822119
```