2d_tests

Vincent Runge

13 avril 2020

Installation of the package from github

```
#devtools::install_github("vrunge/nnlm")
library(nnlm)
```

Data generation

We chose parameters: columns X1 and X2

```
nbSimu <- 10^5
n <- 500
sigma <- 1
X1 <- runif(n, min=0, max = 1)
X2 <- runif(n, min=0, max = 4)

#X1 <- rnorm(n)
#X2 <- rnorm(n)
#X1 <- runif(n, min=0, max = 1)
#X2 <- runif(n, min=0, max = 1)
```

Proba estimation

```
#X1 <- X1 - mean(X1)
#X2 <- X2 - mean(X2)

p <- probaEstimation(X1, X2, sigma, n, nbSimu = nbSimu)
p
```

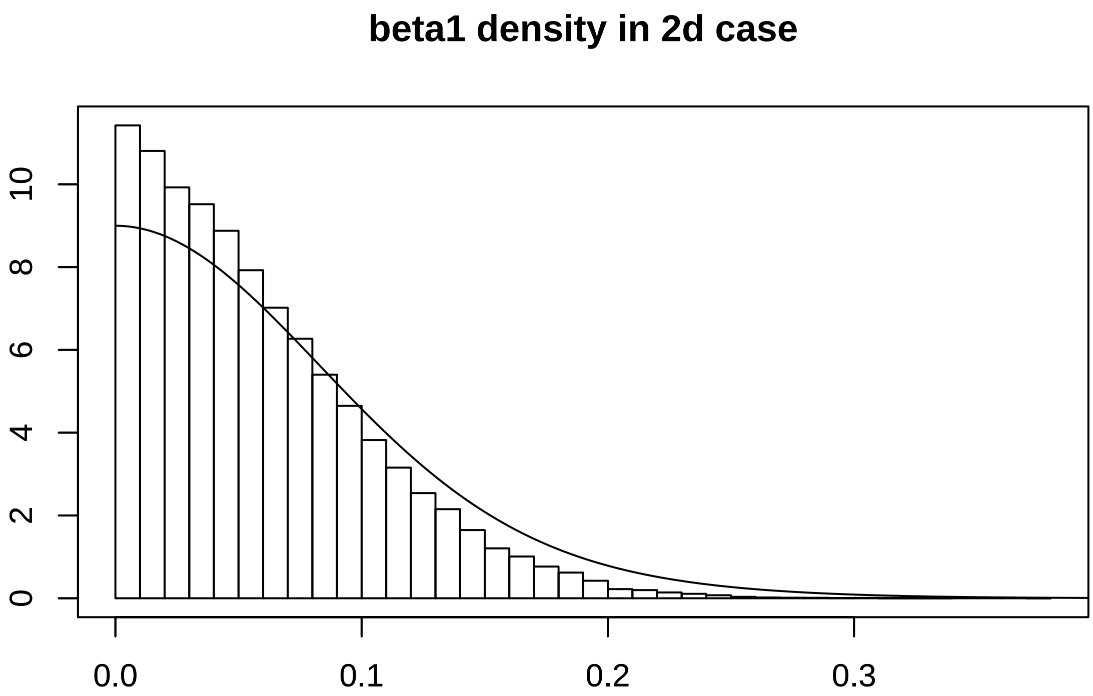
```
## [1] 0.11912 0.37999 0.25113 0.24976
```

Comparison simulation VS estimator

```
step <- 0.001
h <- densitybeta1(X1, X2, sigma, p, h = step, max = 30)
b1 <- beta1estimator0(X1, X2, sigma, n, nbSimu = nbSimu)
```

Plots and tests

```
his <- hist(b1$beta1[b1$beta1>0], breaks = 50, probability = TRUE, ylab = "", xlab = "", main = "")
par(new=TRUE)
plot(h$x,h$y/sum(step*h$y), type = 'l', xlim = c(0,max(his$breaks)), ylim = c(0, his$density[1]), ylab =
```



```
#verif density = 1
his$density%*%diff(his$breaks)
```

```
##      [,1]
## [1,]    1
```

```
sum(step*h$y) + h$d0
```

```
## [1] 1.001667
```

```
#comparaison dirac en 0
sum(b1$beta1 == 0)/nbSimu
```

```
## [1] 0.62928
```

```
p[2] + p[3]
```

```
## [1] 0.63112
```

```
X1carre <- c(X1%*%X1)
X2carre <- c(X2%*%X2)
X12 <- c(X1%*%X2)
Delta <- X1carre*X2carre - X12^2
X12
```

```
## [1] 484.1251
```

```
sqrt(X2carre/Delta)
```

```
## [1] 0.117509
```

```
p[1]
```

```
## [1] 0.11912
```

```
1/sqrt(X1carre)
```

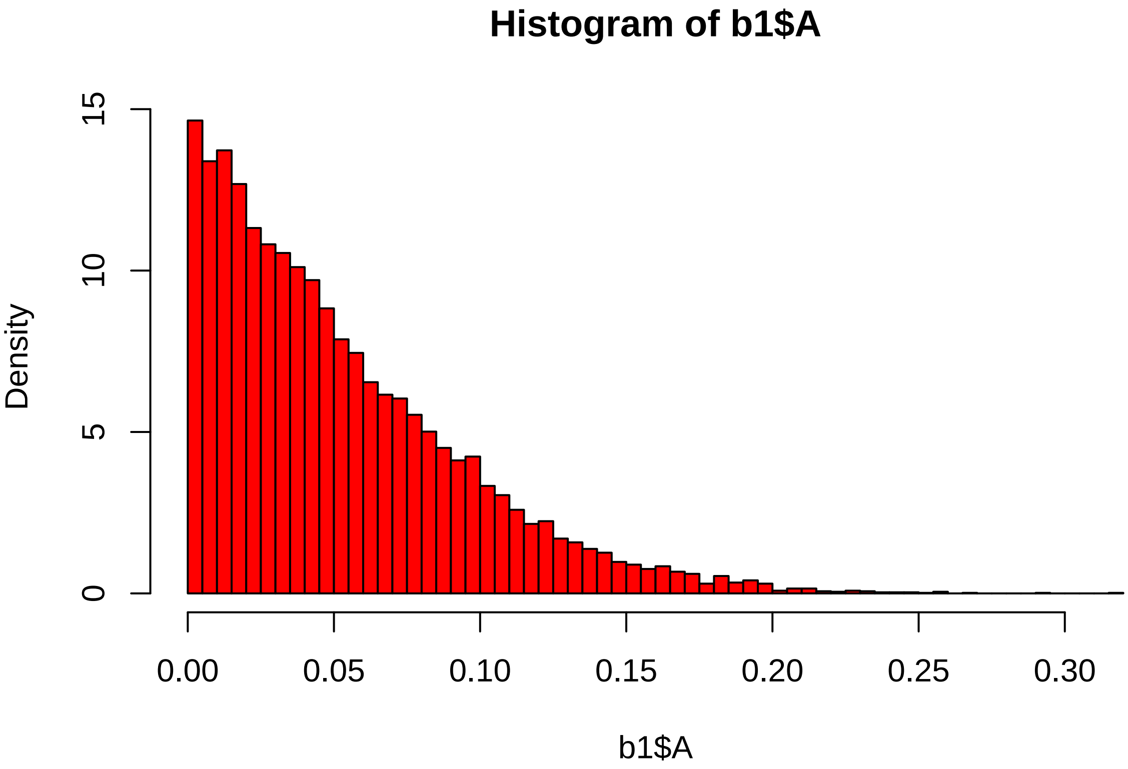
```
## [1] 0.07890569
```

```
p[4]
```

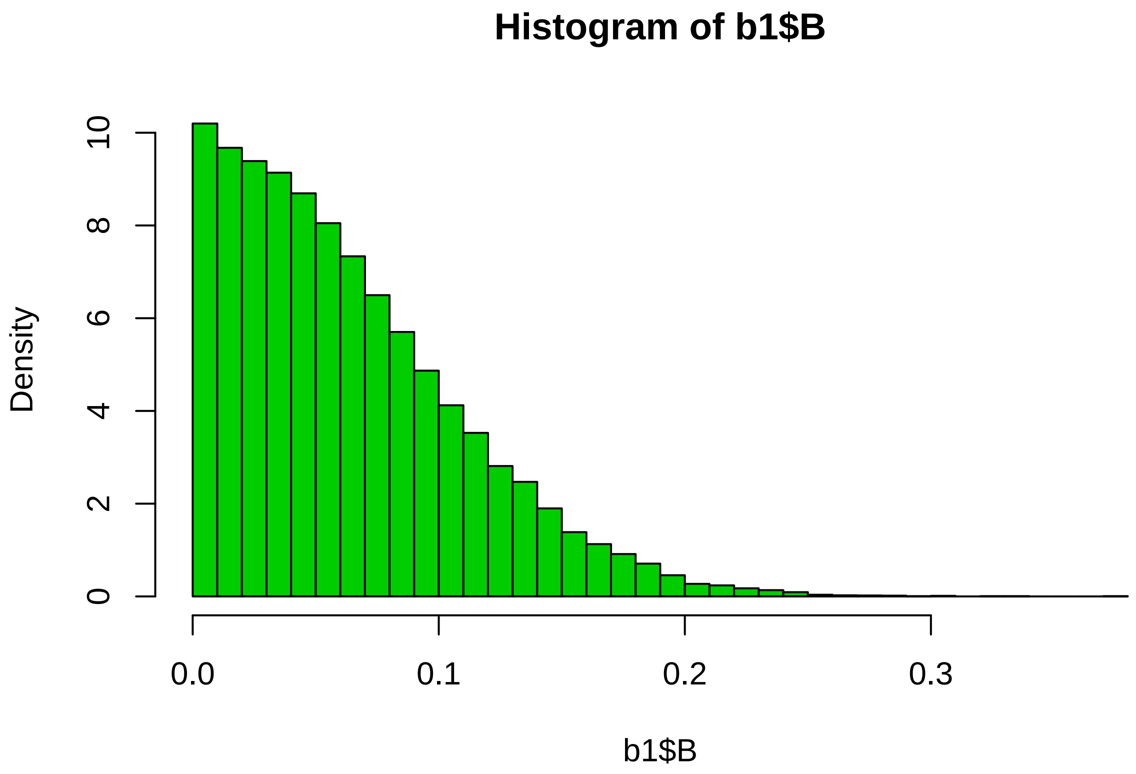
```
## [1] 0.24976
```

Variables A1 and A2 : distributions

```
hist(b1$A, breaks = 50, probability = TRUE, col = 2)
```



```
hist(b1$B, breaks = 50, probability = TRUE, col = 3)
```



```
library(stats)
shapiro.test(sample(c(b1$A,-b1$A), 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(c(b1$A, -b1$A), 5000)
## W = 0.99648, p-value = 1.626e-09
```

```
shapiro.test(sample(c(b1$B,-b1$B), 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(c(b1$B, -b1$B), 5000)
## W = 0.99975, p-value = 0.8427
```

```
shapiro.test(sample(c(b1$A,-b1$A,b1$B,-b1$B), 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(c(b1$A, -b1$A, b1$B, -b1$B), 5000)
## W = 0.99932, p-value = 0.05431
```