Simulation - Modèle marche aléatoire plus bruit

Simulations pour phi = 0

```
rm(list = ls())
```

Installation du package

```
devtools::install_github("aminaghoul/ARRW")
```

```
##
   checking for file ‘/tmp/RtmprMMrb5/remotes44da605bcea0/aminaghoul-ARRW-6136afa/DESCRIPTION’ ...

v  checking for file ‘/tmp/RtmprMMrb5/remotes44da605bcea0/aminaghoul-ARRW-6136afa/DESCRIPTION’ (399ms)
##

-  preparing ‘ARRW’:
##

   checking DESCRIPTION meta-information ...

v  checking DESCRIPTION meta-information
##

-  checking for LF line-endings in source and make files and shell scripts
##

-  checking for empty or unneeded directories
##

-  building ‘ARRW_0.1.0.tar.gz’
##


##
```

```
#devtools::install_github("gtromano/DeCAFS", force = TRUE)
```

Chargement des packages

```
library(DeCAFS)
library(ARRW, quietly = TRUE)
library(tidyverse, quietly = TRUE)
library("gridExtra")
```

Générer le signal avec DeCAFS

On prend comme paramètres :

```
n <- 500
phi <- 0
```

```
sdEta <- 0.8
sdNu <- 0.7
```

```
Y <- dataRWAR(n = n, poisParam = .01, meanGap = 15, phi = phi, sdEta = sdEta, sdNu = sdNu)
```

```
signal <- data.frame(1:n,Y$signal)
colnames(signal) <- c("i", "mu")
signal_decafs <- ggplot(signal) + ggtitle("Signal généré par DeCAFS") + geom_point(aes(x = i ,y = mu))
```

Estimateur du signal théorique

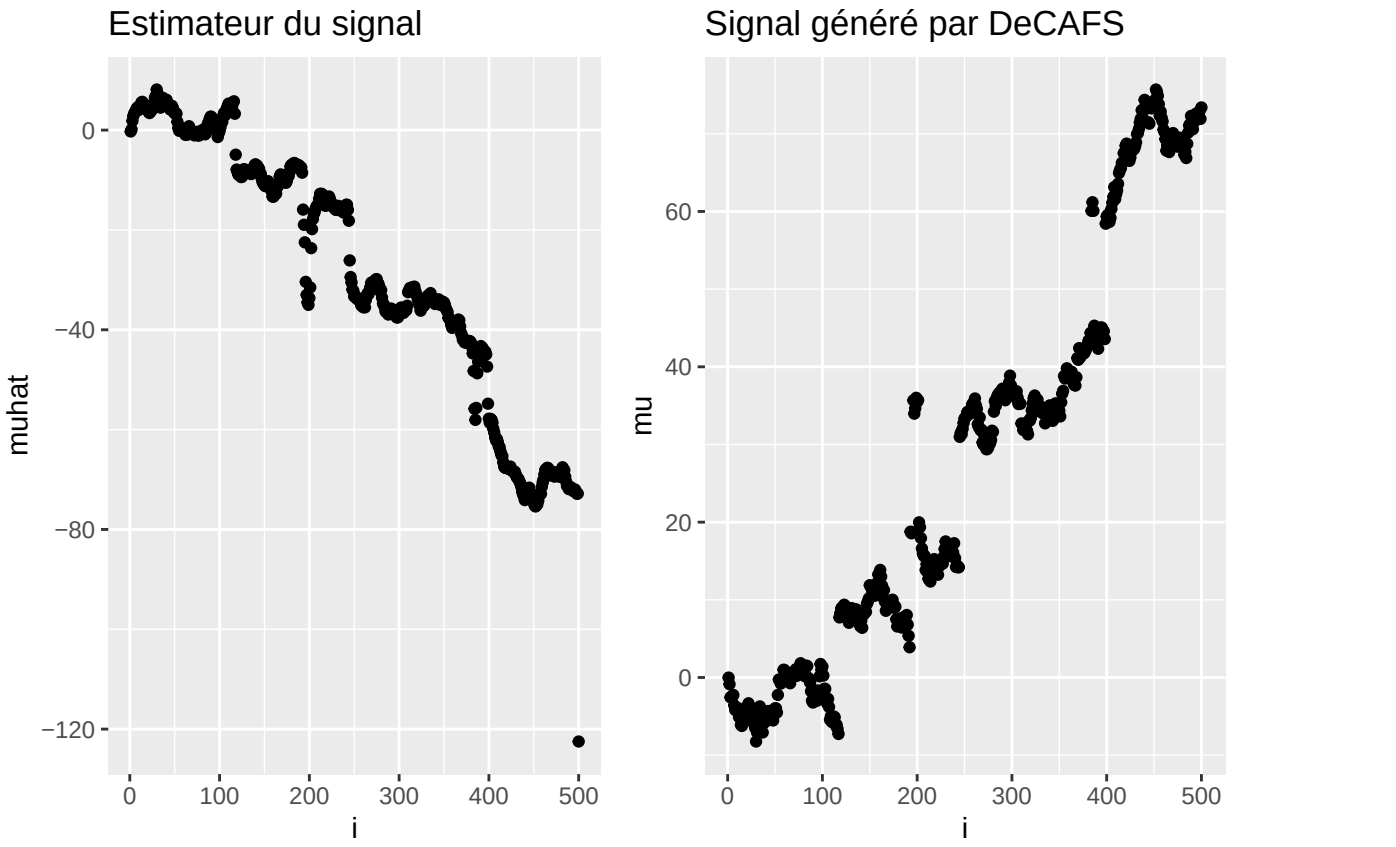
On définit les paramètres suivants :

```
val <- def(sdEta = sdEta, sdNu = sdNu, phi = phi)
kis <- ki(val, n)
omega <- val$om
```

```
estim <- muhat(y = Y$y, kis = kis, omega)
estimateur <- data.frame(1:n,estim)
colnames(estimateur) <- c("i", "muhat")
signal_calc <- ggplot(estimateur) + ggtitle("Estimateur du signal") + geom_point(aes(x = i ,y = muhat))
```

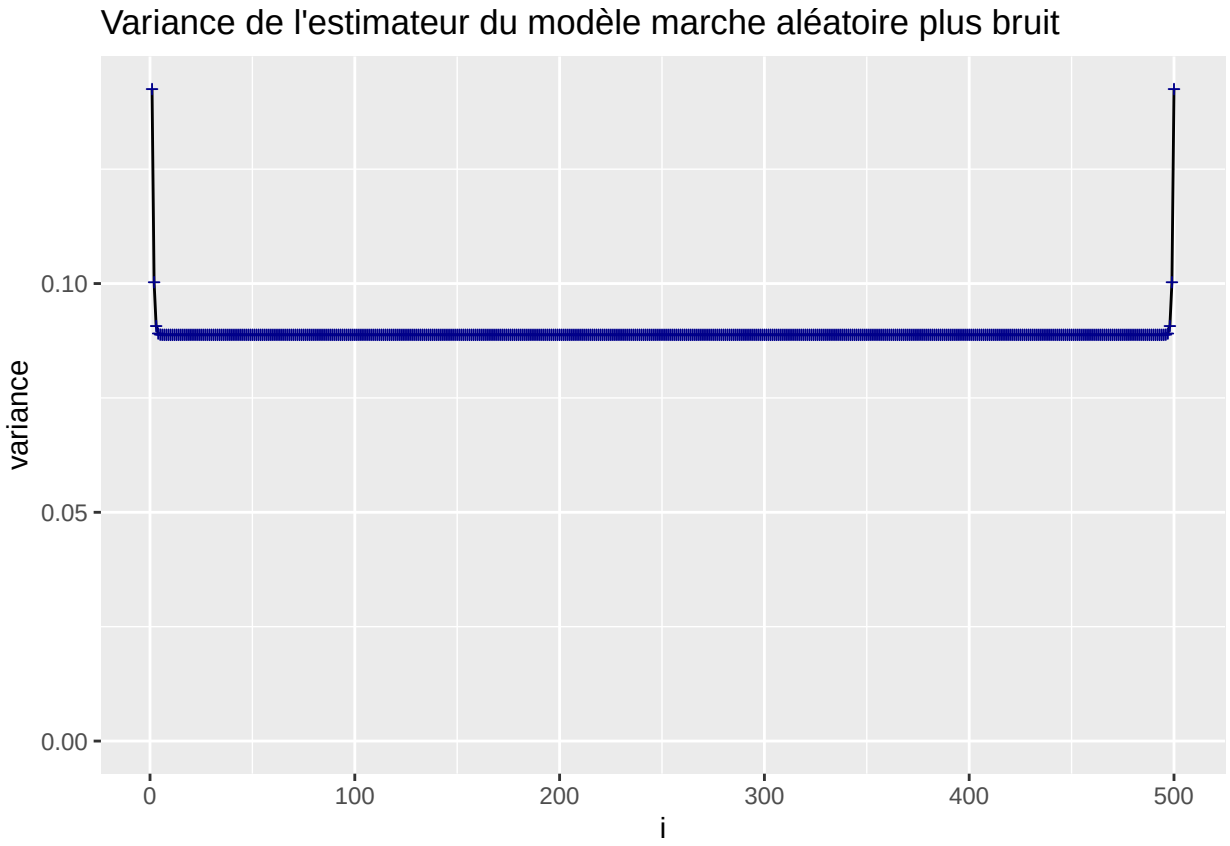
On a alors les deux signaux suivants :

```
grid.arrange(signal_calc, signal_decafs, ncol=2, nrow = 1)
```



La variance de l'estimateur du signal

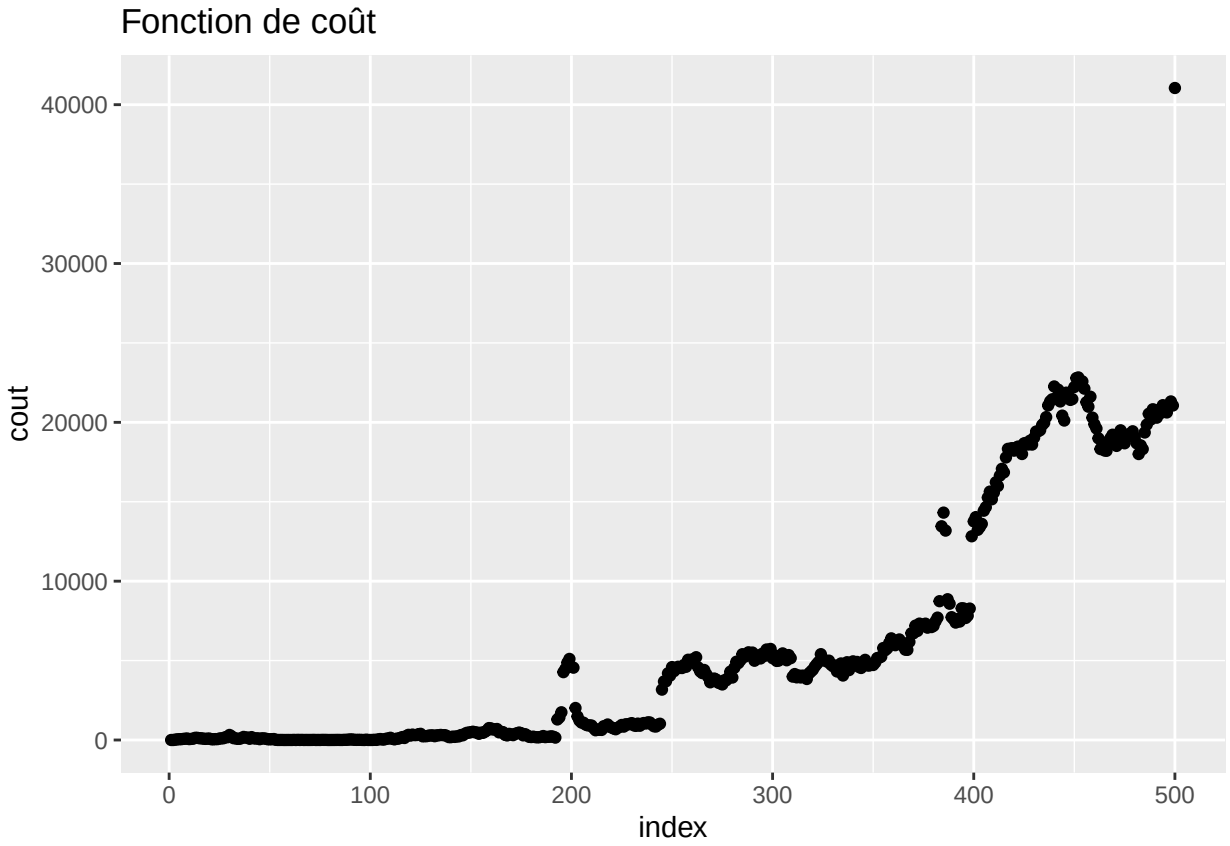
```
res1 <- var1(val, sdNu, 500)
res1 <- data.frame(1:n, res1)
colnames(res1) <- c("i", "variance")
maxi <- max(res1$variance)
ggplot(res1,aes(x = i, y = variance) ) + ggtitle("Variance de l'estimateur du modèle marche aléatoire pl
```



La fonction coût

```
cost <- cost(y = Y$y, estim )
```

```
cout <- data.frame(1:n, cost)
colnames(cout) <- c("index", "cout")
ggplot(cout) + ggtitle("Fonction de coût") + geom_point(aes(x = index , y = cout))
```


Fonction de coût
40000
30000
20000
10000
0
cout
0
100
200
300
400
500
index