

# Simulation - Modèle marche aléatoire plus bruit

## Simulations pour $\phi = 0$

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

## Installation du package

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

## Chargement des packages

```
library(DeCAFS)  
library(ARRW, quietly = TRUE)  
# Pour ggplot2 et la manipulation des dataframes  
library(tidyverse, quietly = TRUE)  
# Pour tracer plusieurs graphiques en même temps  
library(gridExtra, quietly = TRUE)
```

## Générer le signal avec DeCAFS

On prend comme paramètres :

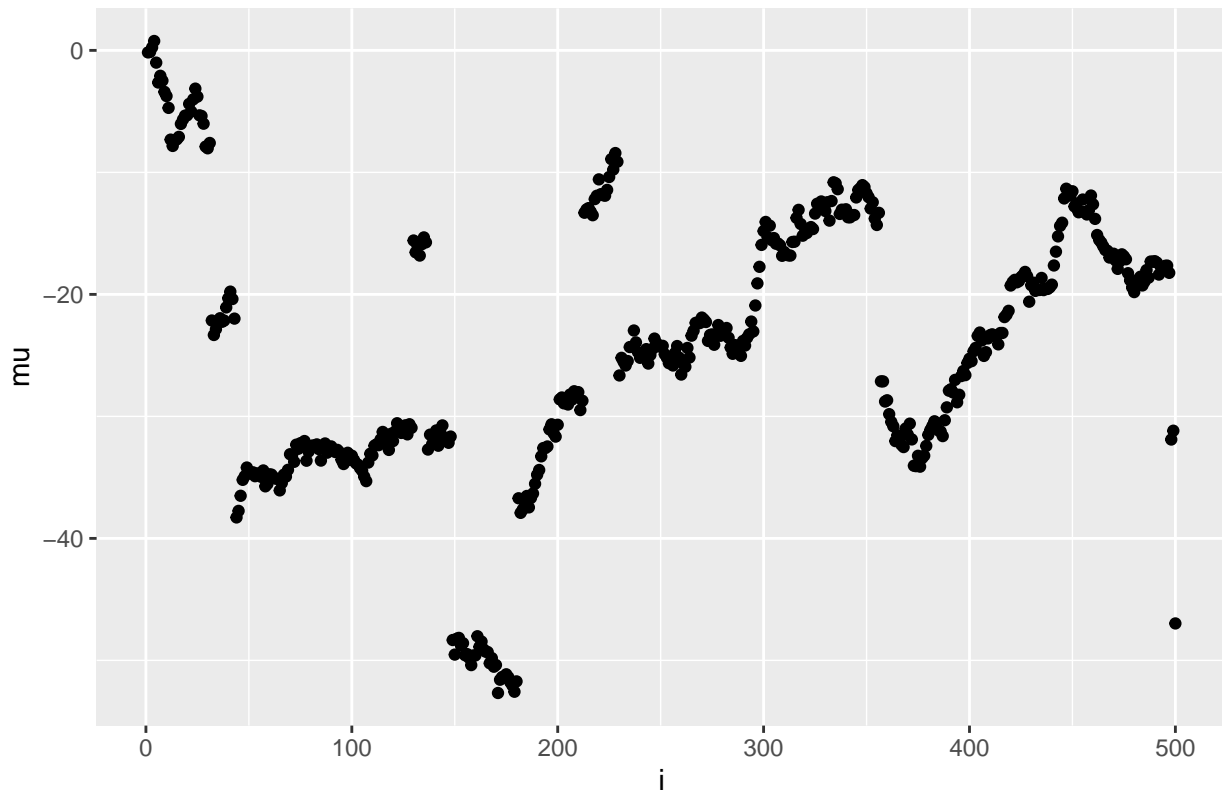
```
n <- 500  
phi <- 0  
sdEta <- 0.8  
sdNu <- 0.7
```

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

On trace le signal généré.

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

## Signal généré par DeCAFS



## Estimateur du signal théorique

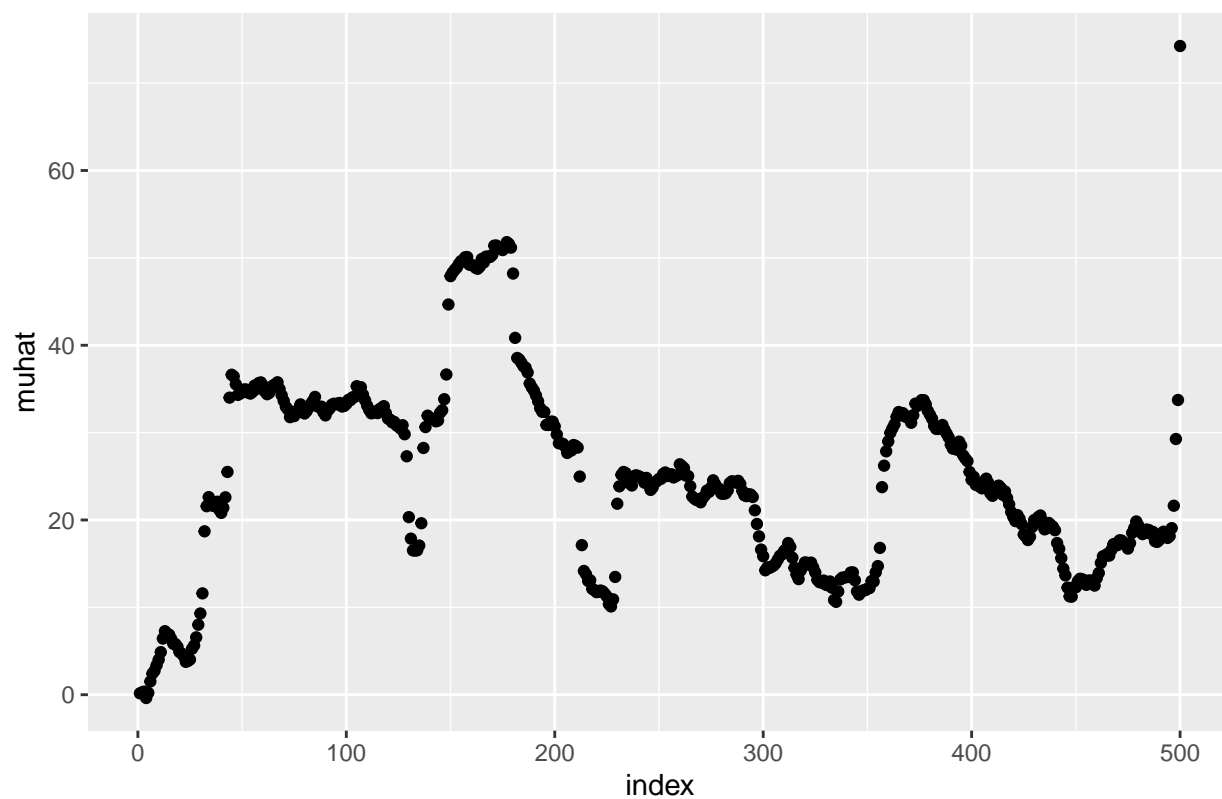
On définit les paramètres suivants :

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

On trace l'estimateur du signal

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

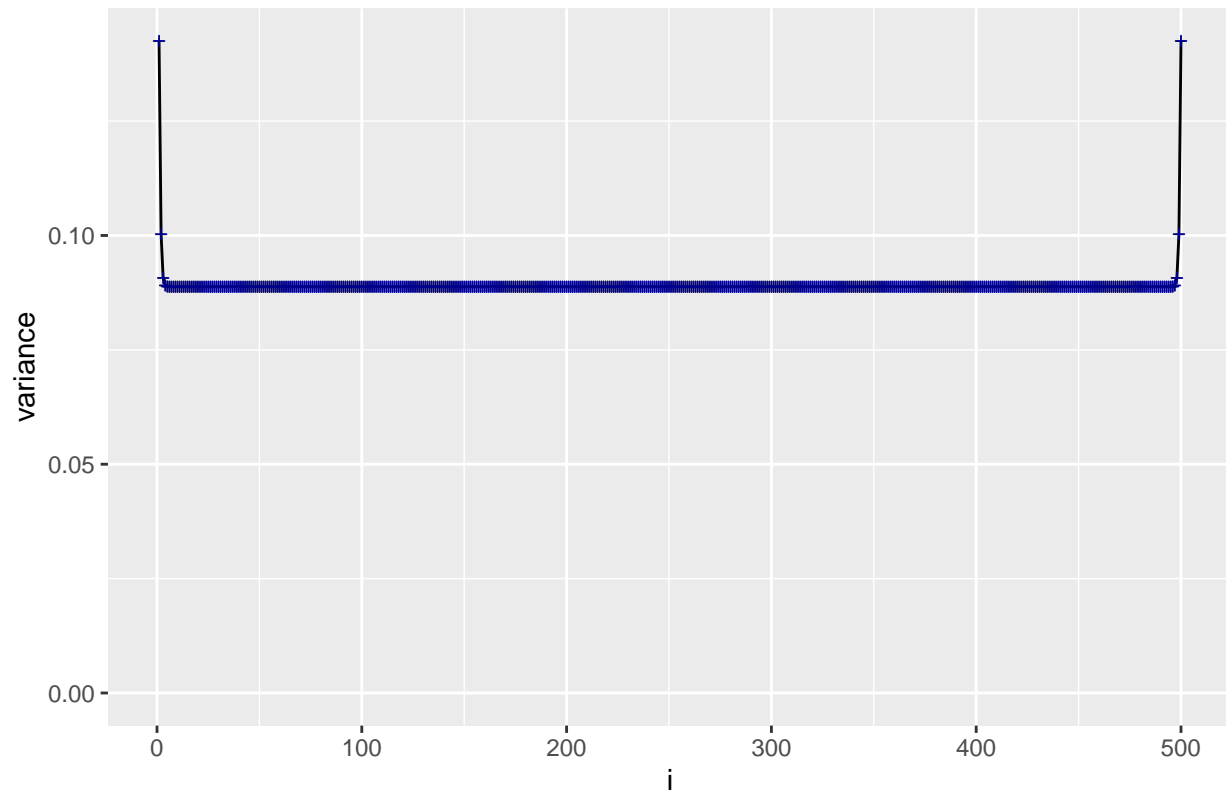
## Estimateur du signal



## La variance de l'estimateur du signal

```
res1 <- var1(val, sdNu, 500)
maxi <- max(res1$variance)
ggplot(res1, aes(x = i, y = variance)) + ggtitle("Variance de l'estimateur du modèle marche aléatoire p
```

### Variance de l'estimateur du modèle marche aléatoire plus bruit



### 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))
```

