Méthode des rectangles

Christopher Blier-Wong

November 23, 2017

Résumé

Ce document présente la méthode des rectangles. Il est basé sur un document préparé par Jérémie Boudreault pour le dépannage en ACT-3000 lors de la session A16.

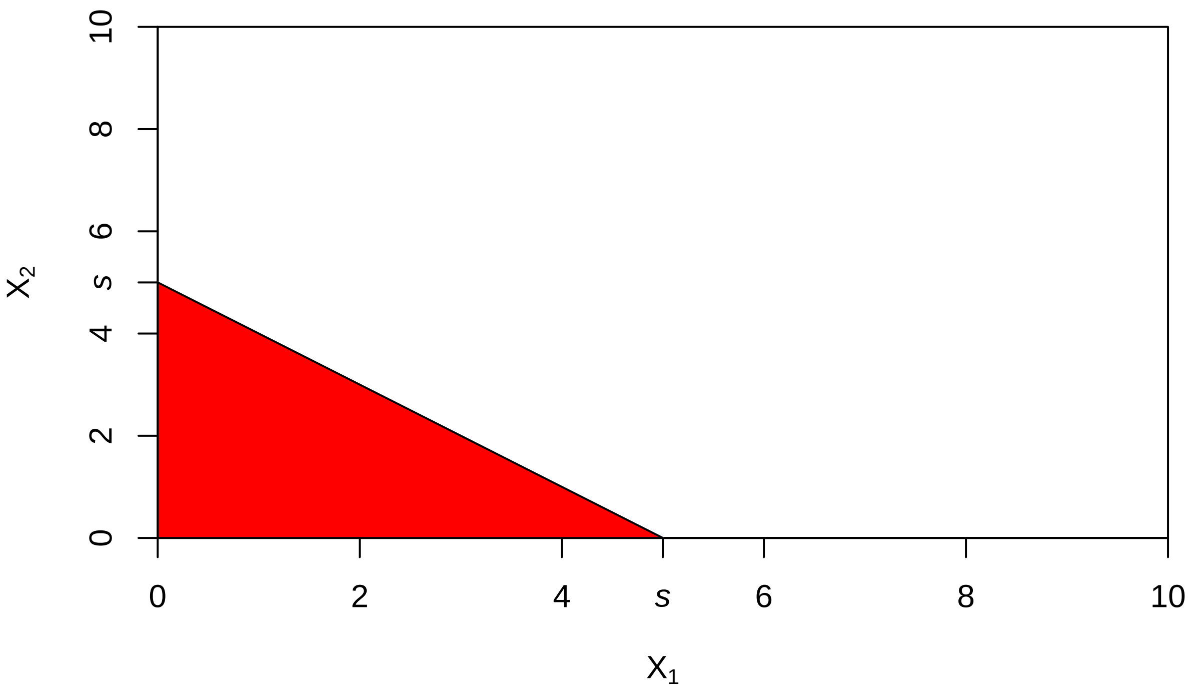
Méthode des rectangles

Soit $S = X_1 + X_2$. On s'intéresse à la fonction de répartition de S.

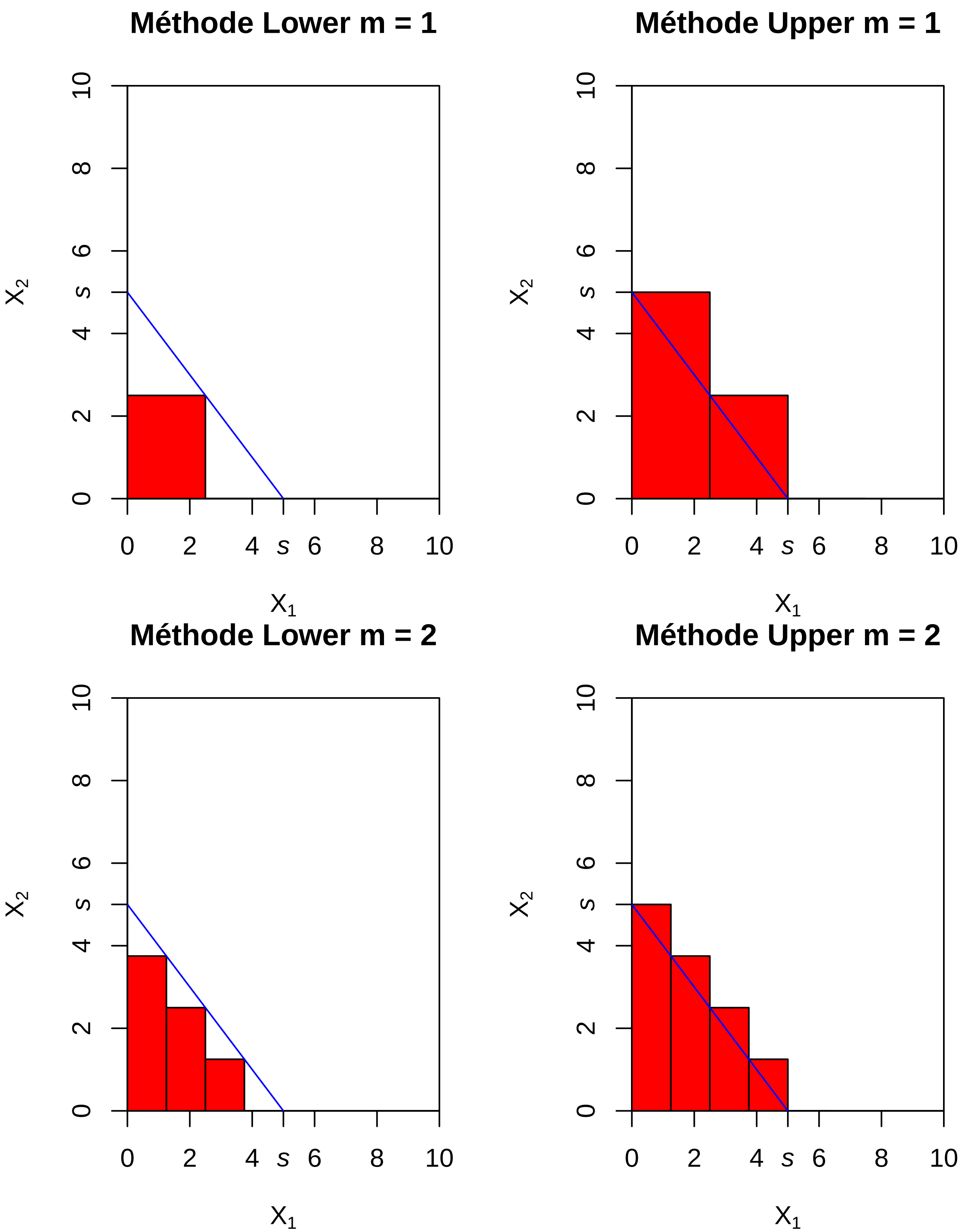
On a

$$\begin{aligned} Pr(S \leq s) &= Pr(X_1 + X_2 \leq s) \\ &= Pr(X_2 \leq s - X_1). \end{aligned}$$

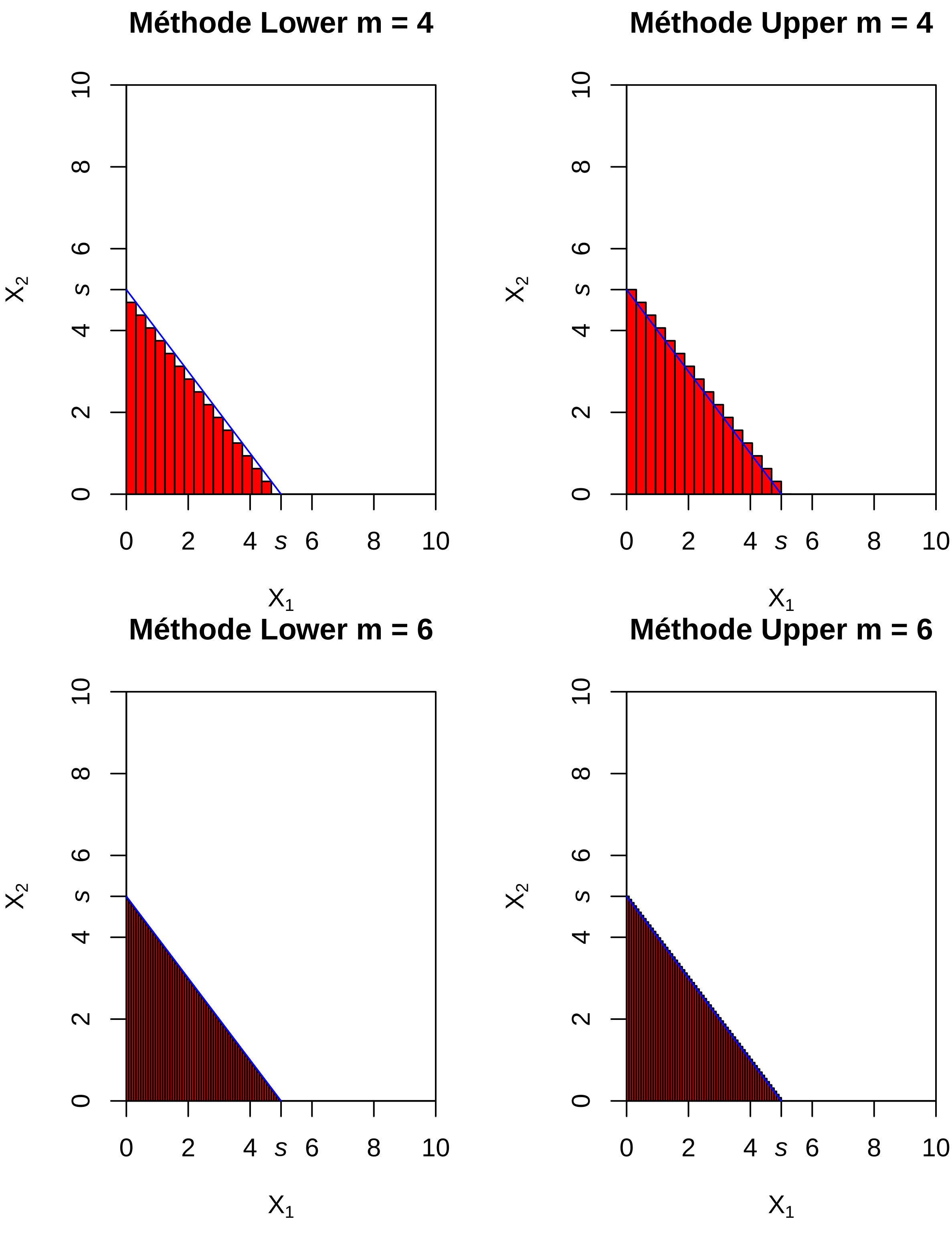
On trace le graphique de $X_2 \leq s - X_1$ et on veut calculer l'aire sous le graphique.



On obtient un triangle. Par contre, on a seulement la fonction de répartition bivariée, qui forme un rectangle dans le graphique. La méthode des rectangles peut approximer le triangle par plusieurs rectangles, comme suit:



Méthode Lower m = 1
Méthode Upper m = 1
Méthode Lower m = 2
Méthode Upper m = 2
X_1
X_2
0 2 4 s 6 8 10



Pour faire l'estimation de la fonction de répartition, on sépare l'intervalle $[0, s]$ en 2^m régions. La méthode lower sous-estime la fonction de répartition (et sur-estime la VaR, sur-estime l'espérance). On remarque qu'il faut calculer $2^m - 1$ rectangles. La notation de l'estimation de $F_S(s)$ par la méthode lower est $A_S^{(l,m)}$. La méthode upper sur-estime la fonction de répartition (et sous-estime la VaR, sous-estime l'espérance). On remarque qu'il faut calculer 2^m rectangles. La notation de l'estimation de $F_S(s)$ par la méthode upper est

$A_S^{(u,m)}$. Dans les deux cas, on a

$$\lim_{m\to\infty} A_S^{(l,m)} = \lim_{m\to\infty} A_S^{(u,m)} = F_S(s).$$

Retour sur la question 10.1.1.4.iii.

```r
DensiteCopule <- function(u1, u2) {
  u1 + u2 - 1 + ((1 - u1) ** (- 2) + (1 - u2) ** (- 2) - 1) ** (- 1 / 2)
}

Fy1 <- function(y1) {
  1 - (20 / (20 + y1)) ** 3
}

Fy2 <- function(y2) {
  pgamma(y2, shape = 2, rate = 1 / 10)
}

Fy1y2 <- function(y1, y2) {
  DensiteCopule(Fy1(y1), Fy2(y2))
}

CalcuerRepartition <- function(m, s){
  pas <- s / (2 ** m)
  ylower <- seq(pas, s - pas, by = pas)
  yupper <- seq(pas, s, by = pas)
  a.lower <- sum(sapply(ylower, function(t) Fy1y2(t, s - t) - Fy1y2(t - pas, s - t)))
  a.upper <- sum(sapply(yupper, function(t) Fy1y2(t, s - t + pas) - Fy1y2(t - pas, s - t  + pas)))
  print(paste0("Lower: ", a.lower, ". Upper: ",  a.upper, "."))
}

CalcuerRepartition(1, 60)
```

```
## [1] "Lower: 0.797896040200001. Upper: 0.938307502837249."
```

```r
CalcuerRepartition(2, 60)
```

```
## [1] "Lower: 0.856927464937274. Upper: 0.924685208225902."
```

```r
CalcuerRepartition(3, 60)
```

```
## [1] "Lower: 0.882803425986913. Upper: 0.911703897404423."
```

```r
CalcuerRepartition(4, 60)
```

```
## [1] "Lower: 0.891247806365642. Upper: 0.905631696902639."
```

```r
CalcuerRepartition(5, 60)
```

```
## [1] "Lower: 0.89514678986317. Upper: 0.90232509258287."
```

```r
CalcuerRepartition(6, 60)
```

```
## [1] "Lower: 0.897016134806785. Upper: 0.900603583631087."
```

```r
CalcuerRepartition(10, 60)
```

```
## [1] "Lower: 0.89872230081888. Upper: 0.898946481050581."
```