# Analyse spatiale des Arbres du POSL

Florence Puech

Eric Marcon

February 14, 2022

**Abstract**

Représentations spatiales du jeu de données et dbmss.

Ce code crée des cartes en 2D et 3D du Parc Omnisport Suzanne Lenglen.

## 1 Données

```
load("data/POSL.RData")
```

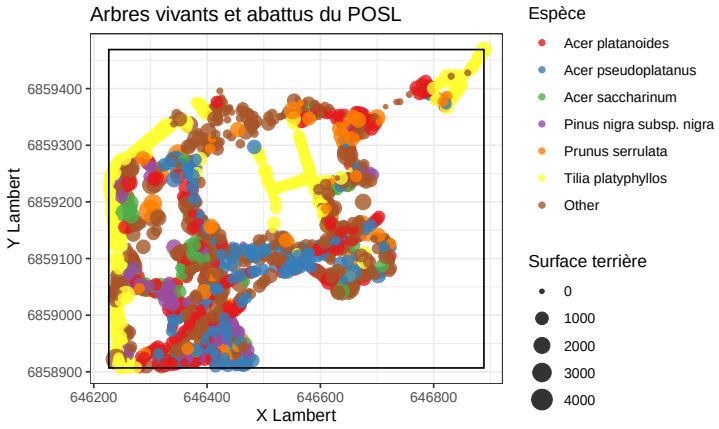
## 2 Jeux de points

### 2.1 Genre-Espèce

```
library("dbmss")
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    rename(PointType = GenrEsp) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    POSL_VA_wmppp
```

#### 2.1.1 Carte

```
POSL_VA_wmppp %>%
    autoplot(alpha = 0.8, xlab = "X Lambert", ylab = "Y Lambert") +
    labs(title = "Arbres vivants et abattus du POSL") +
    labs(color = "Espèce", size = "Surface terrière")
```

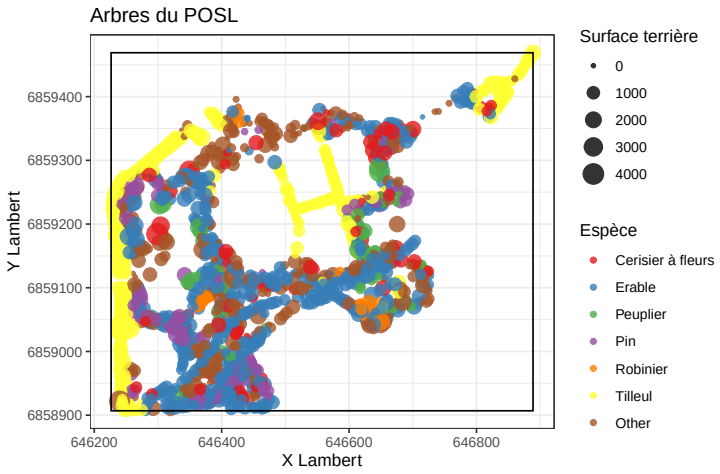



## 2.2 Nom vernaculaire des espèces

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    rename(PointType = EspeceFrancais) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    POSL_Esp_wmppp
```

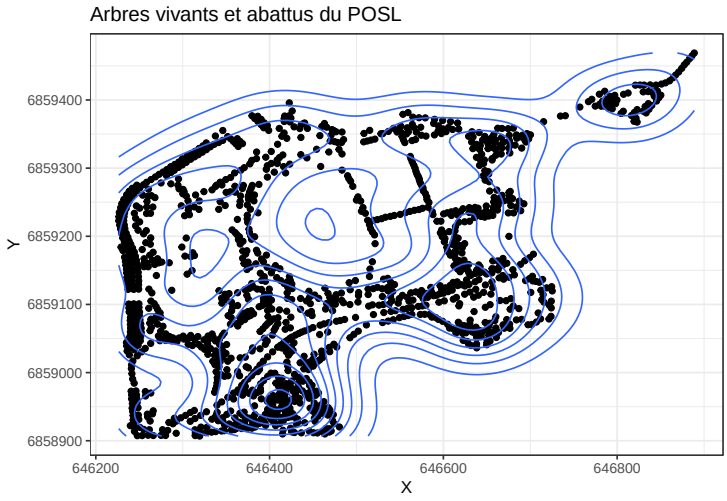
### 2.2.1 Carte

```
POSL_Esp_wmppp %>%
    autoplot(alpha = 0.8, xlab = "X Lambert", ylab = "Y Lambert") +
    labs(title = "Arbres du POSL") + labs(color = "Espèce",
    size = "Surface terrière")
```




### 2.2.2 Vue en 2D

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    ggplot(aes(x = X, y = Y)) + geom_point() + geom_density_2d() +
    ggtitle("Arbres vivants et abattus du POSL")
```

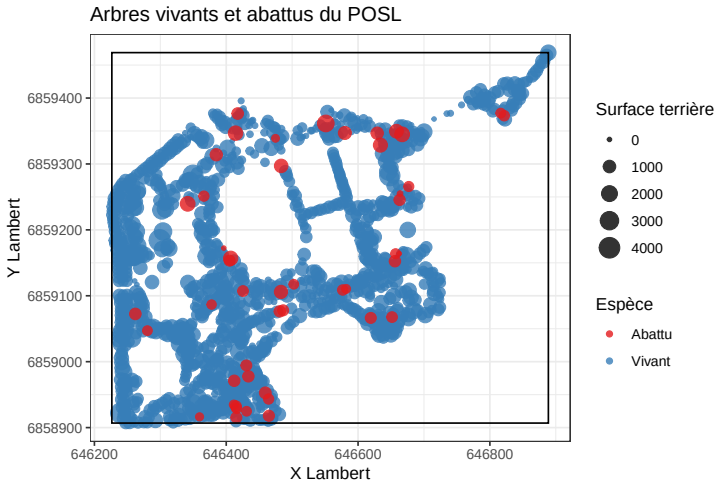



## 2.3 Abattus-vivants

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    rename(PointType = Etat) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    BDD_V2022_A2022_Poids_POSL_wmppp
```

### Carte

```
BDD_V2022_A2022_Poids_POSL_wmppp %>%
    autoplot(alpha = 0.8, xlab = "X Lambert", ylab = "Y Lambert") +
    labs(title = "Arbres vivants et abattus du POSL") +
    labs(color = "Espèce", size = "Surface terrière")
```

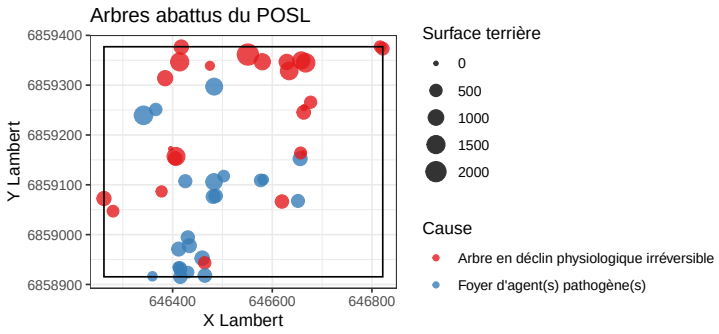



## 2.4 Cause de l'abattage

```
# Création du jeu de points
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Arbres abattus seulement
filter(Etat == "Abattu") %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    rename(PointType = MotifAbattage) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    POSL_A_wmppp
```

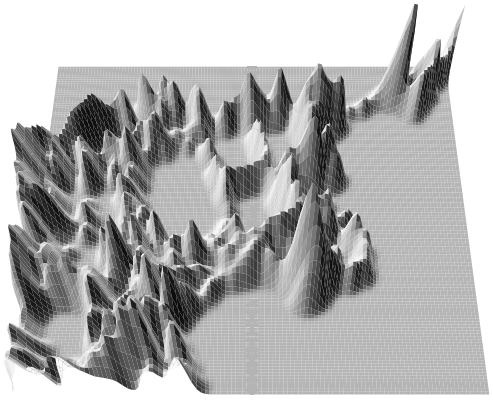
### Carte

```
POSL_A_wmppp %>%
    autoplot(alpha = 0.8, xlab = "X Lambert", ylab = "Y Lambert") +
    labs(title = "Arbres abattus du POSL") + labs(color = "Cause",
    size = "Surface terrière")
```




# 3 Carte 3D densité arbres vivants et abattus à POSL

```
Density_POSL <- density(POSL_VA_wmppp, bw.diggle(POSL_VA_wmppp),
    dimyx = c(128, 128))
par(mar = c(0, 0, 0, 0))
persp.im(Density_POSL, shade = 0.5, scale = FALSE,
    axes = TRUE, border = NA, box = FALSE, phi = 60,
    main = "") -> Projection
```



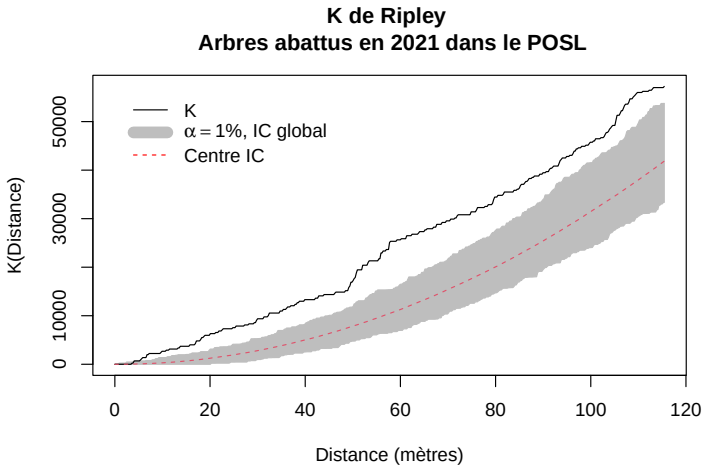
# 4 Concentration spatiale

## 4.1 Question 1-a : les arbres abattus du POSL sont-ils plus concentrés qu'une distribution aléatoire ?

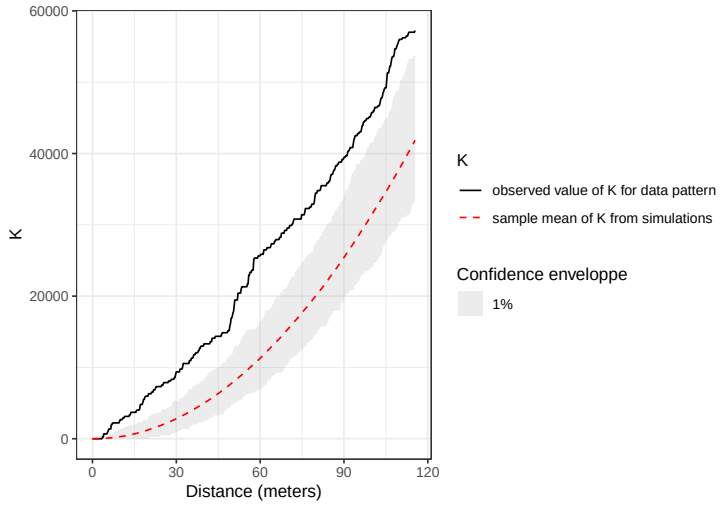
```
K_Abattus <- KEnvelope(POSL_A_wmppp, NumberOfSimulations = 1000,
    Alpha = 0.01, Global = TRUE)
```

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(K_Abattus, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "K(Distance)", main = "K de Ripley \n Arbres abattus en 2021 dans le POSL")
legend("topleft", c("K", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.04)
```



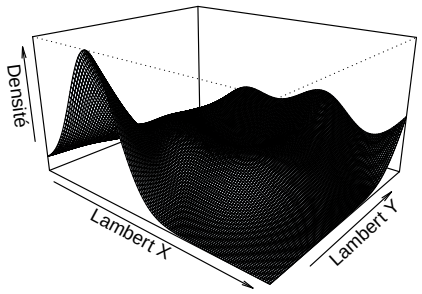

```
autoplot(K_Abattus)
```




Conclusion : interactions détectée entre les arbres abattus : attraction. Les arbres abattus sont plus concentrés qu'une distribution complètement aléatoire.

```
persp(density(POSL_A_wmppp),
      col="aliceblue", # couleur triste = arbres abattus
      theta = 40, phi = 20,
      xlab = "Lambert X", ylab = "Lambert Y", zlab = "Densité",
      main = "Densité arbres abattus en 2021 dans le POSL")
```

**Densité arbres abattus en 2021 dans le POSL**



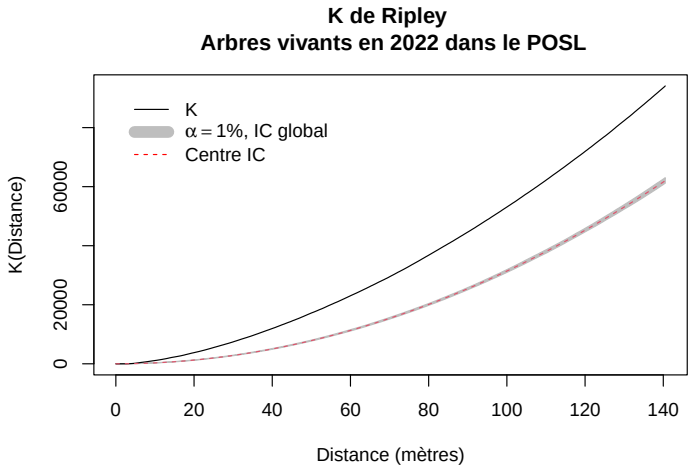

## 4.2 Question 1-b : les arbres vivants du POSL sont-ils plus concentrés qu'une distribution aléatoire ?

```
K_Vivants <- KEnvelope(POSL_Esp_wmppp, NumberOfSimulations = 1000,
    Alpha = 0.01, Global = TRUE)
```

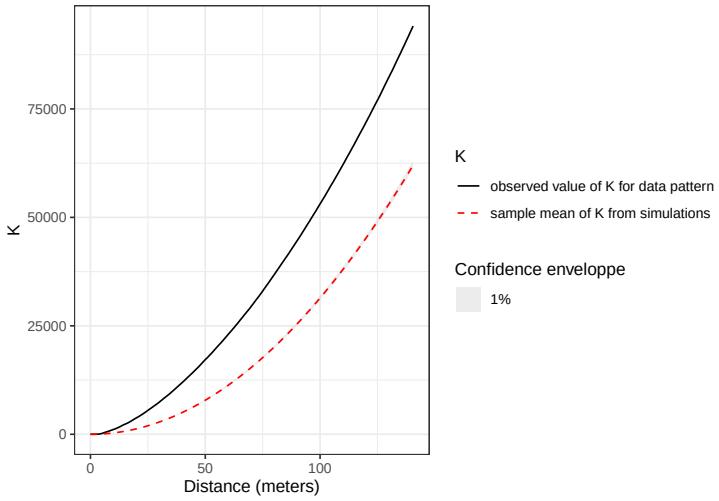
```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(K_Vivants, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "K(Distance)", main = "K de Ripley \n Arbres vivants en 2022 dans le POSL")
legend("topleft", c("K", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
```

```
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.04)
```




```
autoplot(K_Vivants)
```




Conclusion : interactions détectée entre les arbres vivants : attraction. Les arbres vivants sont plus concentrés qu'une distribution complètement aléatoire.

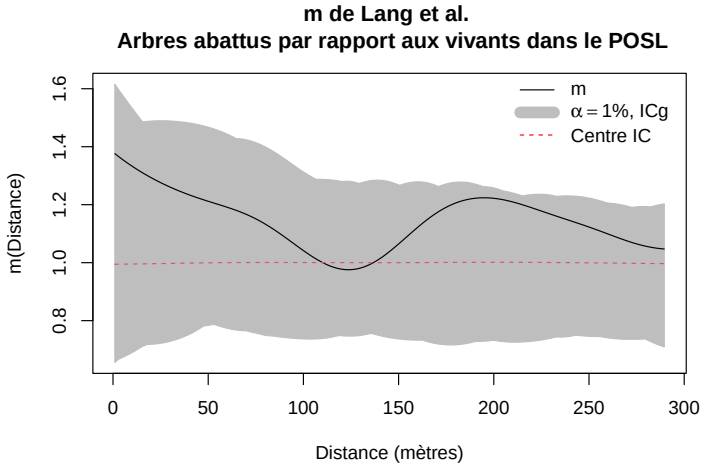
## 4.3 Question 2 : les arbres abattus du POSL sont-ils plus concentrés que les vivants ?

### 4.3.1 mCas_contrôles pour les abattus

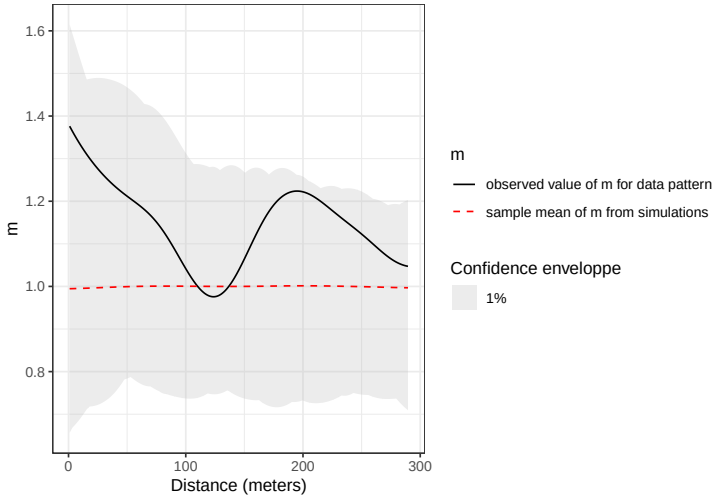
```
m_Abattus <- mEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Abattu",
    SimulationType = "RandomLocation", Global = TRUE)
```

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(m_Abattus, xlab = "Distance (mètres)", ylab = "m(Distance)",
    legend = FALSE, main = "m de Lang et al. \n Arbres abattus par rapport aux vivants dans le POSL")
legend("topright", c("m", expression(alpha == "1%, ICg    "),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = -0.02)
```




```
autoplot(m_Abattus)
```




Conclusions :

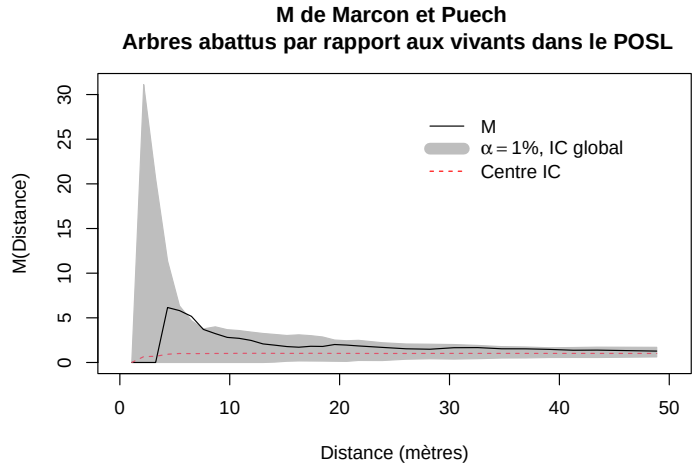
- NS avec alpha 1% et 1000 sim
- avec alpha 10% concentration sign entre 175m et 200m avec 1000 sim, presque plus sign avec 10000 sim.

### 4.3.2 MCas_contrôles pour les abattus

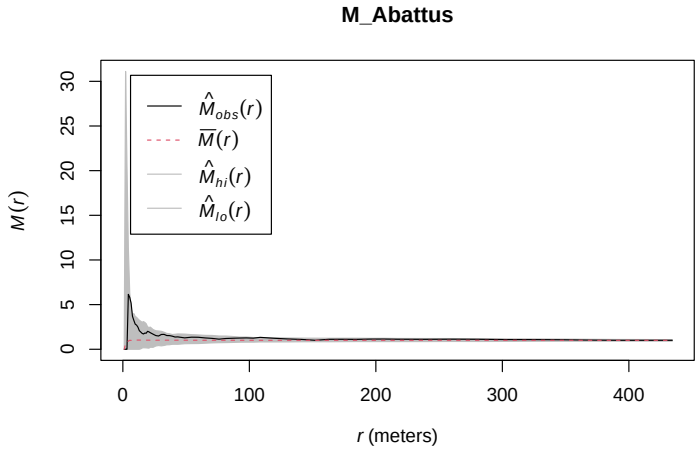
```
M_Abattus <- MEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Abattu",
    SimulationType = "RandomLocation", Global = TRUE)
```

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(M_Abattus, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "M(Distance)", xlim = c(0, 50), main = "M de Marcon et Puech \n Arbres abattus par rapport aux vivants dans le POSL")
legend("topright", c("M", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.1)
```




```
plot(M_Abattus)
```




Conclusion : un peu conc sign autour de 5m.
Attention : attention warning message lors de la compil.

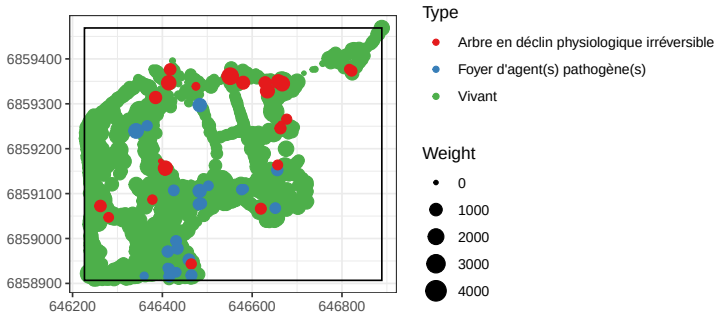
```
Warning messages: 1: In FUN(newX[, i], ...) :
aucun argument trouvé pour min ; Inf est renvoyé"
```

## 4.4 Question 3 : les malades sont-ils plus concentrés ceux en délin par rapport aux vivants ?

### 4.4.1 Question 3-a : REFERENTIEL : ARBRES VIVANTS et ABATTUS POUR AUTRES MOTIFS

Création de la BDD :

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    mutate(PointType = ifelse(Etat == "Vivant", "Vivant",
        as.character(MotifAbattage))) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    BDD_V2022_A2022_Poids_POSL_wmppp2
autoplot(BDD_V2022_A2022_Poids_POSL_wmppp2)
```



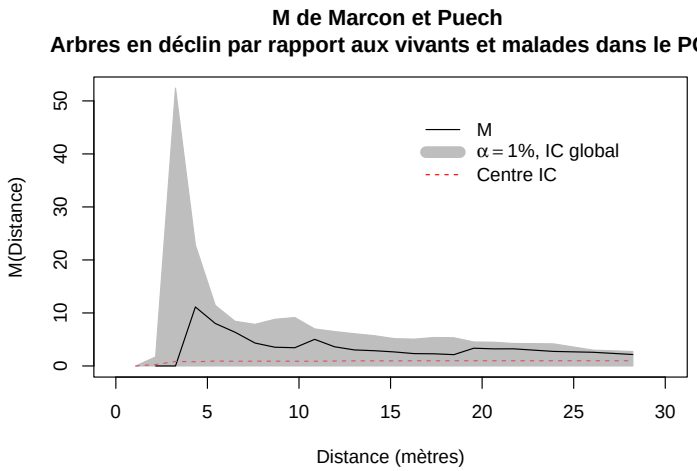

```
M_Declin <- MEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp2,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Arbre en déclin physiologique irréversible",
    SimulationType = "RandomLocation", Global = TRUE)
```

## M déclin

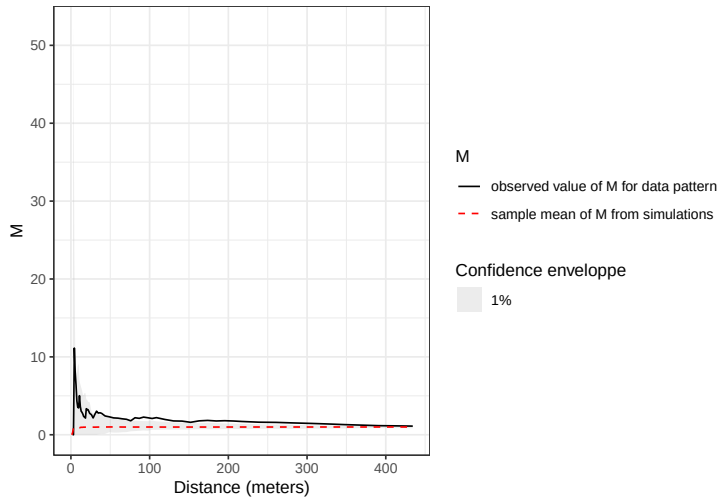
```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(M_Declin, legend = FALSE, xlim = c(0, 30), xlab = "Distance (mètres)",
    ylab = "M(Distance)", main = "M de Marcon et Puech \n Arbres en déclin par rapport aux vivants et malades dans le POSL")
legend("topright", c("M", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
```

```
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.1)
```




```
autoplot(M_Declin)
```



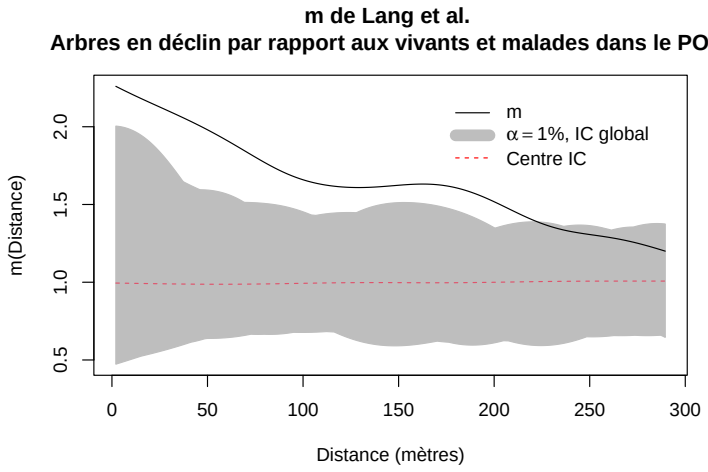

Conclusion : NS.

```
m_Declin <- mEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp2,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Arbre en déclin physiologique irréversible",
    SimulationType = "RandomLocation", Global = TRUE)
```

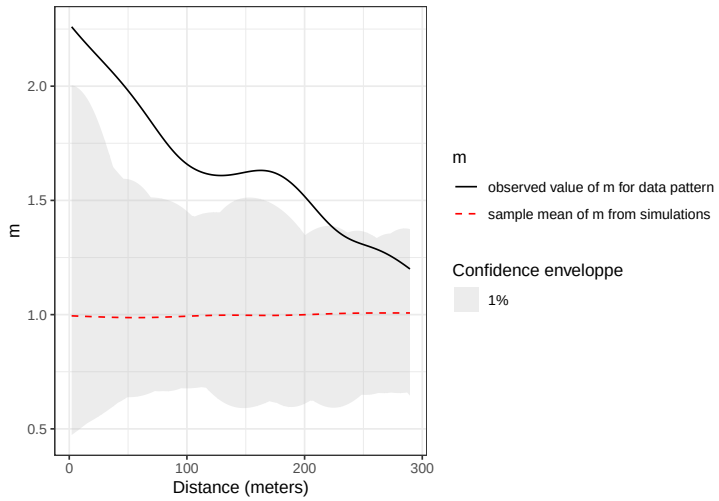
### m déclin

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(m_Declin, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "m(Distance)", main = "m de Lang et al. \n Arbres en déclin par rapport aux vivants et malades dans le POSL")
legend("topright", c("m", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.05)
```




```
autoplot(m_Declin)
```



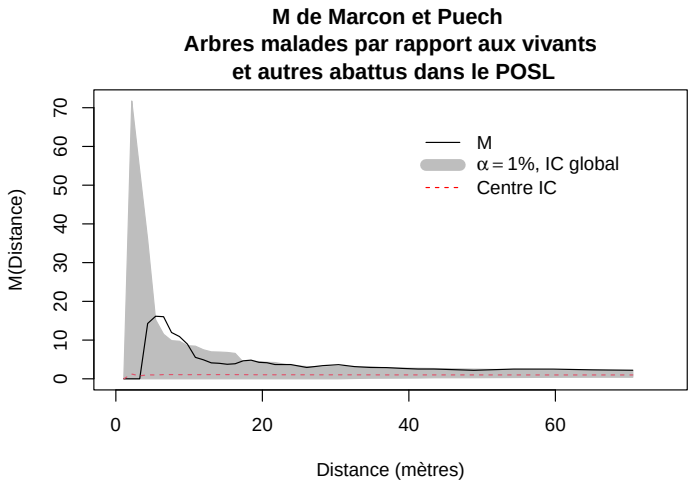

Conclusion : conc sign jusqu'à 100m puis jusqu'à 250m.

```
M_Malades <- MEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp2,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Foyer d'agent(s) pathogène(s)",
    SimulationType = "RandomLocation", Global = TRUE)
```

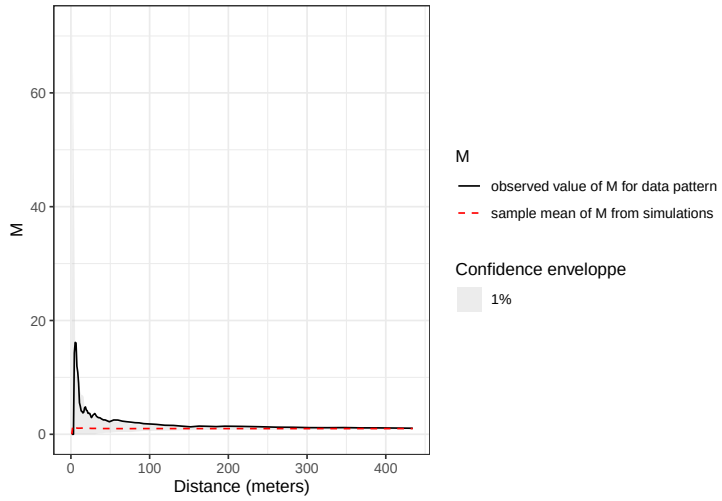
## M malade

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990..... 1000.
##
## Done.
```

```r
plot(M_Malades, legend = FALSE, xlim = c(0, 75), xlab = "Distance (mètres)",
    ylab = "M(Distance)", main = "M de Marcon et Puech \n Arbres malades par rapport aux vivants \n et autres abattus dans le POSL")
legend("topright", c("M", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.1)
```




```r
autoplot(M_Malades)
```



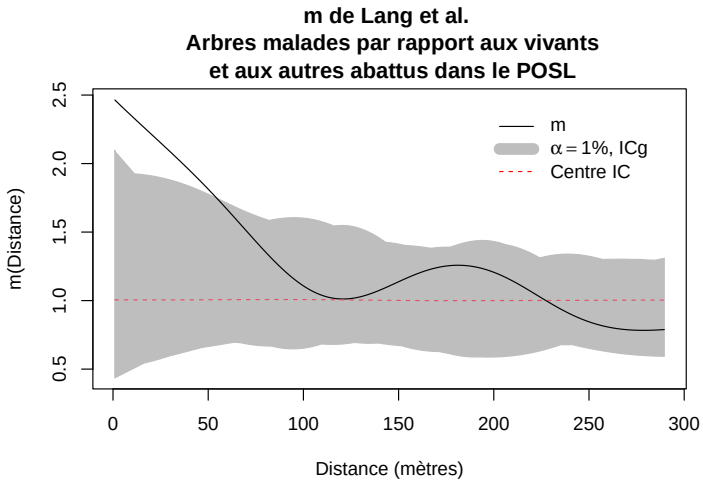

Conclusion : conc à petites distances (environ 10m) Attention : warnings FUN

```
m_Malades <- mEnvelope(BDD_V2022_A2022_Poids_POSL_wmppp2,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Foyer d'agent(s) pathogène(s)",
    SimulationType = "RandomLocation", Global = TRUE)
```

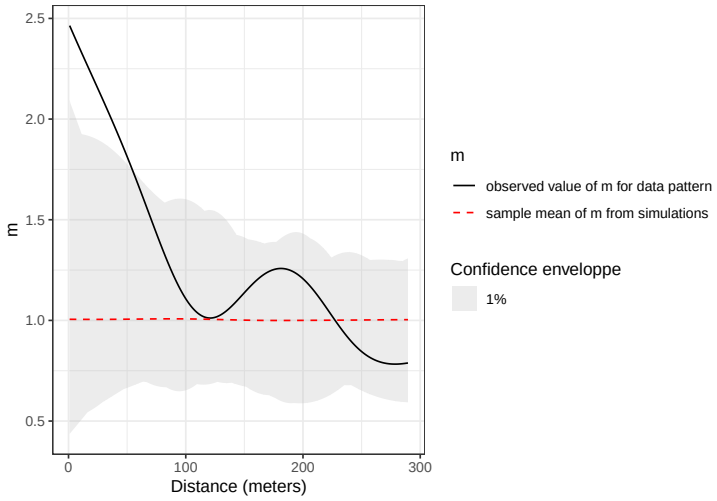
## m malade

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(m_Malades, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "m(Distance)", main = "m de Lang et al. \n Arbres malades par rapport aux vivants\n et aux autres abattus dans le POSL
legend("topright", c("m", expression(alpha == "1%, ICg"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.05)
```



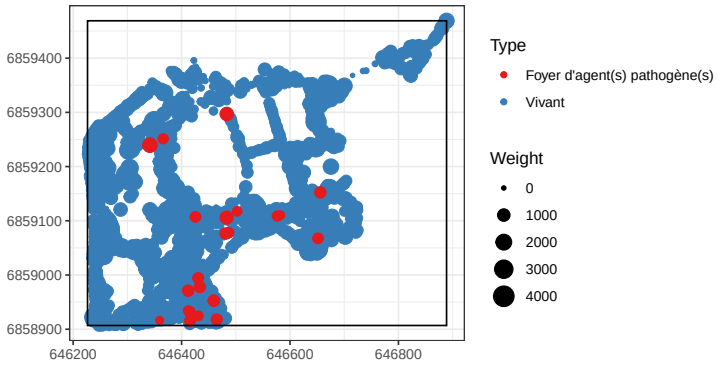
```
autoplot(m_Malades)
```




### 4.4.2 Question 3-b : REFERENTIEL : UNIQUEMENT LES ARBRES VIVANTS

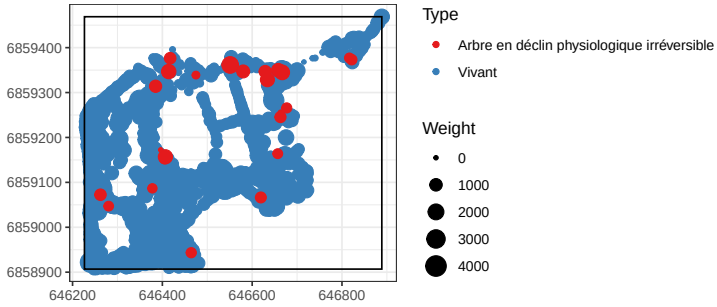
**Jeux de points** Création de la BDD sans les 25 arbres en déclin donc contient vivant et abattus-patho :

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    filter(MotifAbattage != "Arbre en déclin physiologique irréversible") %>%
    mutate(PointType = ifelse(Etat == "Vivant", "Vivant",
        "Foyer d'agent(s) pathogène(s)")) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    BDD_V2022_A2022_Poids_POSL_sans_abattus_declin_wmppp
autoplot(BDD_V2022_A2022_Poids_POSL_sans_abattus_declin_wmppp)
```




Création de la BDD sans les 23 arbres foyer patho donc reste uniquement vivants et en déclin:

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    filter(MotifAbattage != "Foyer d'agent(s) pathogène(s)") %>%
    mutate(PointType = ifelse(Etat == "Vivant", "Vivant",
        "Arbre en déclin physiologique irréversible")) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    BDD_V2022_A2022_Poids_POSL_sans_patho_wmppp
autoplot(BDD_V2022_A2022_Poids_POSL_sans_patho_wmppp)
```

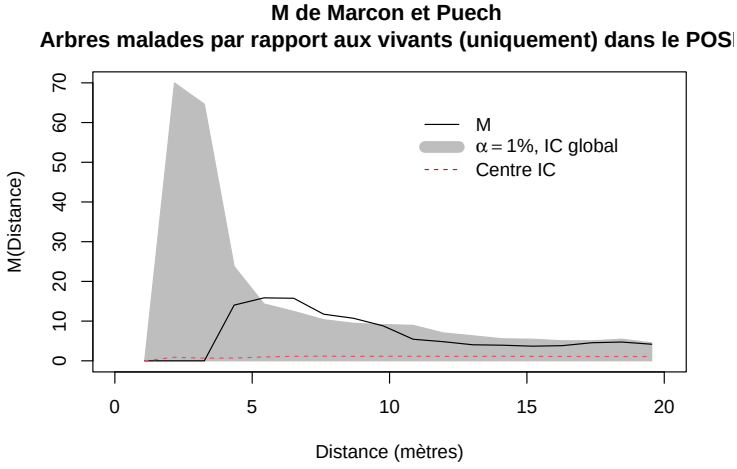


```
M_Malades <- MEnvelope(BDD_V2022_A2022_Poids_POSL_sans_abattus_declin_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Foyer d'agent(s) pathogène(s)",
    SimulationType = "RandomLocation", Global = TRUE)
```

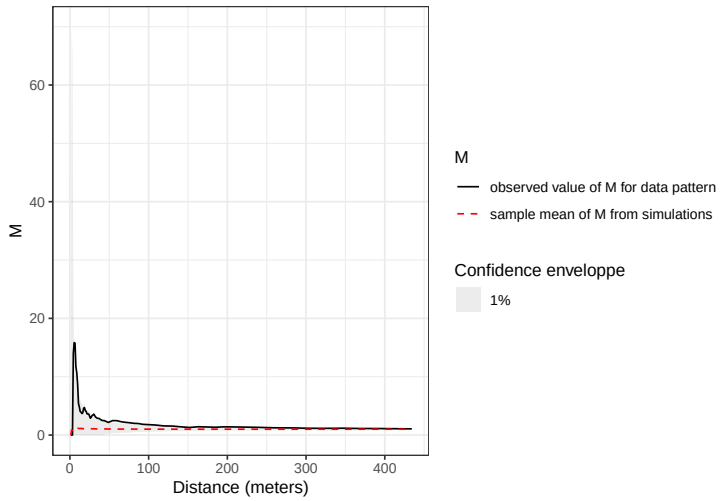
## M malades

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(M_Malades, legend = FALSE, xlim = c(0, 20), xlab = "Distance (mètres)",
    ylab = "M(Distance)", main = "M de Marcon et Puech \n Arbres malades par rapport aux vivants (uniquement) dans le POSL")
legend("topright", c("M", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.1)
```



```
autoplot(M_Malades)
```



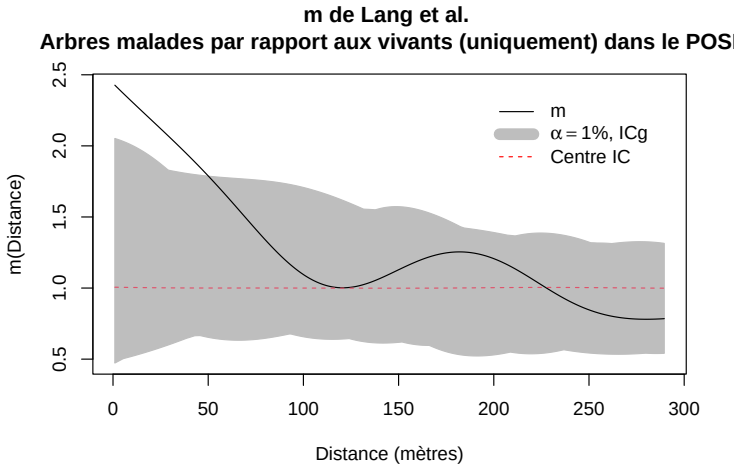

Conclusion : conc sign entre 6m et 8m et M est alors = 10 ou 11 !!! Remarque : warning fun.

```
m_Malades <- mEnvelope(BDD_V2022_A2022_Poids_POSL_sans_abattus_declin_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Foyer d'agent(s) pathogène(s)",
    SimulationType = "RandomLocation", Global = TRUE)
```

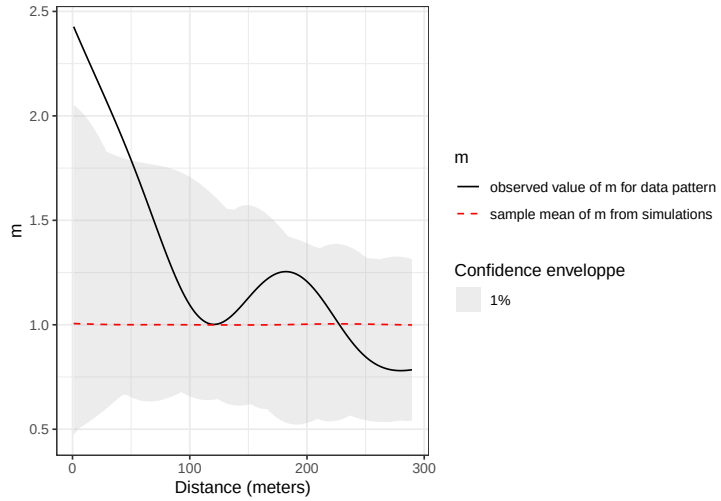
## m malades

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(m_Malades, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "m(Distance)", main = "m de Lang et al. \n Arbres malades par rapport aux vivants (uniquement) dans le POSL")
legend("topright", c("m", expression(alpha == "1%, ICg"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.05)
```




```
autoplot(m_Malades)
```

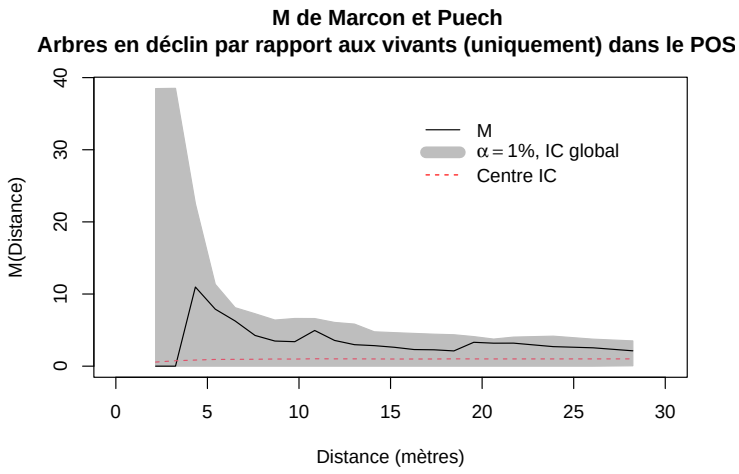



```
M_Declin <- MEnvelope(BDD_V2022_A2022_Poids_POSL_sans_patho_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Arbre en déclin physiologique irréversible",
    SimulationType = "RandomLocation", Global = TRUE)
```

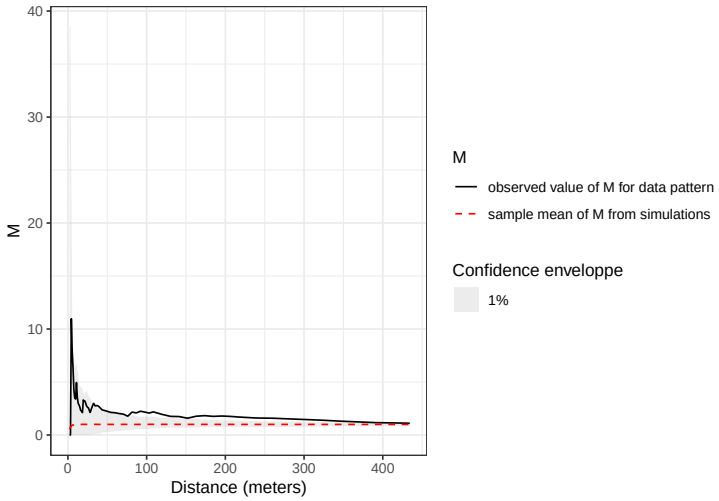
## M déclin

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(M_Declin, legend = FALSE, xlim = c(0, 30), xlab = "Distance (mètres)",
    ylab = "M(Distance)", main = "M de Marcon et Puech \n Arbres en déclin par rapport aux vivants (uniquement) dans le POSL")
legend("topright", c("M", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.1)
```




```
autoplot(M_Declin)
```



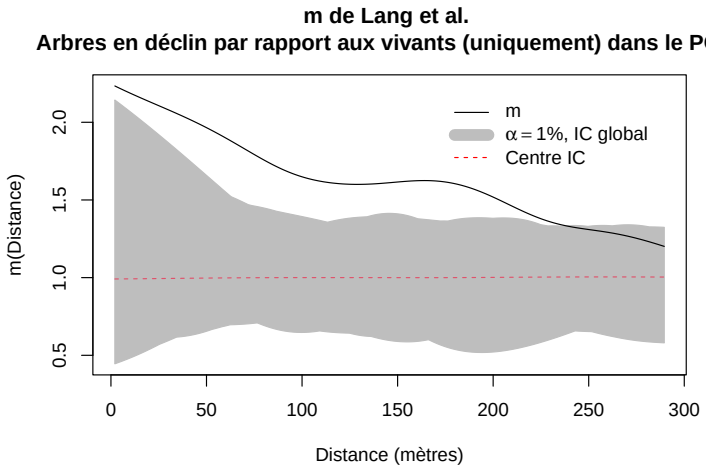

```
m_Declin <- mEnvelope(BDD_V2022_A2022_Poids_POSL_sans_patho_wmppp,
    NumberOfSimulations = 1000, Alpha = 0.01, ReferenceType = "Arbre en déclin physiologique irréversible",
    SimulationType = "RandomLocation", Global = TRUE)
```

## m déclin

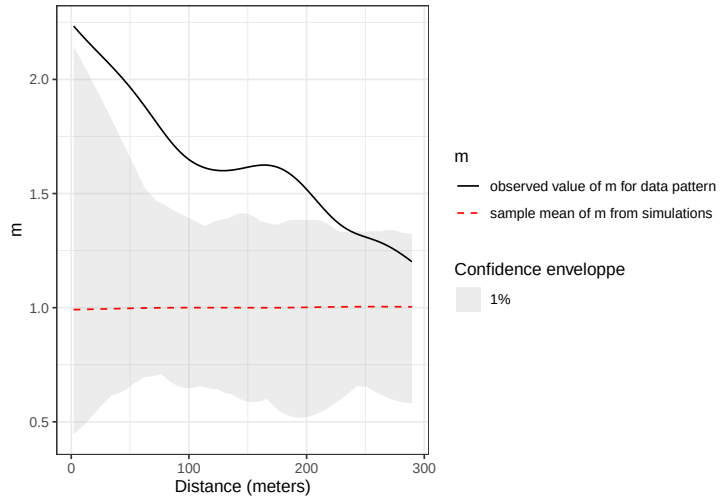
```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(m_Declin, legend = FALSE, xlab = "Distance (mètres)",
    ylab = "m(Distance)", main = "m de Lang et al. \n Arbres en déclin par rapport aux vivants (uniquement) dans le POSL")
legend("topright", c("m", expression(alpha == "1%, IC global"),
    "Centre IC"), col = c("black", "grey", "red"),
```

```
    lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
    text.col = "black", horiz = FALSE, inset = 0.05)
```



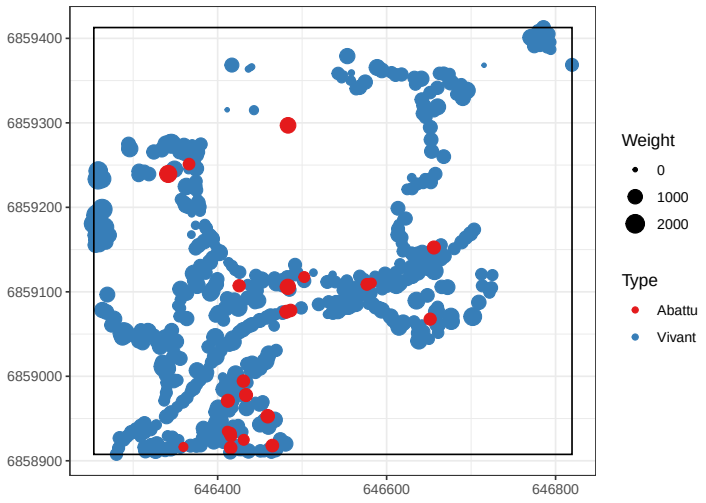

```
autoplot(m_Declin)
```




## 4.5 Question 4 : les érables malades sont-ils plus concentrés que les érables vivants ?

### 4.5.1 Jeu de points

```
BDD_Vivants_et_Abattus_renseignes_POSL %>%
    filter(EspeceFrancais == "Erable") %>%
    # Le poids est la surface terrière
mutate(PointWeight = Circonference^2/4/pi) %>%
    filter(MotifAbattage != "Arbre en déclin physiologique irréversible") %>%
    mutate(PointType = ifelse(Etat == "Vivant", "Vivant",
        "Abattu")) %>%
    as.wmppp(unitname = c("meter", "meters")) ->
    BDD_V2022_A2022_Poids_POSL_erables_VPatho_wmppp
autoplot(BDD_V2022_A2022_Poids_POSL_erables_VPatho_wmppp)
```



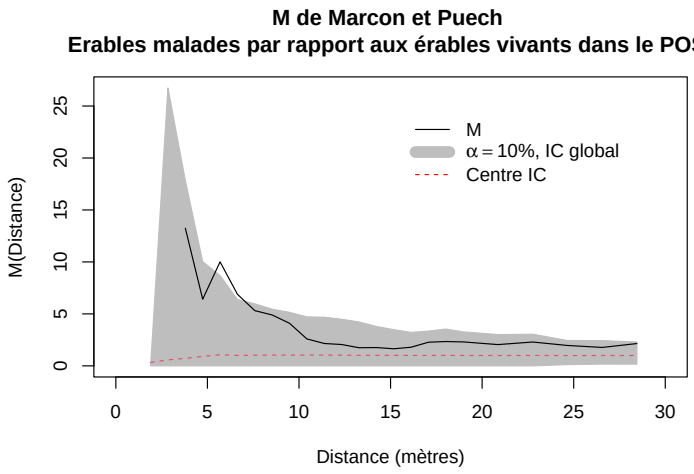

```
M_erables <- MEnvelope(BDD_V2022_A2022_Poids_POSL_erables_VPatho_wmppp,
                       NumberOfSimulations=1000,
                       Alpha=0.1,  # attention : 10% ici
                       ReferenceType = "Abattu",
                       SimulationType = "RandomLocation",
                       Global=TRUE)
```

## M érables foyer patho

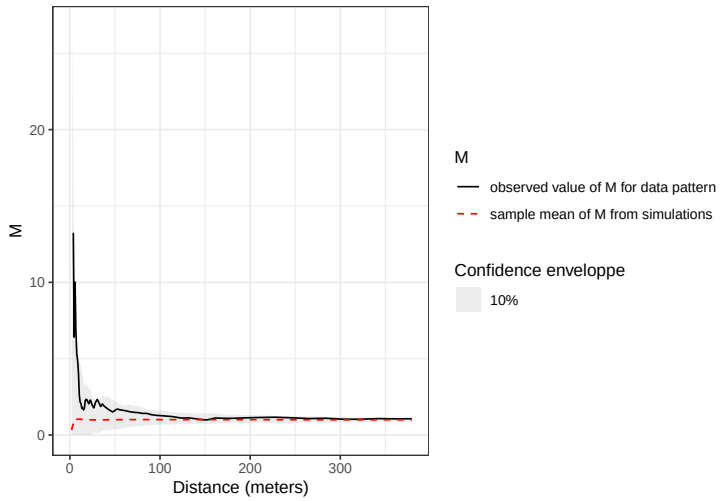
```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
```

```
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```
plot(M_erables,
     legend = FALSE,
     xlim=c(0,30),
     xlab="Distance (mètres)", ylab="M(Distance)",
     main="M de Marcon et Puech \n Erables malades par rapport aux érables vivants dans le POSL")
legend("topright",c("M", expression(alpha=='10%, IC global'), "Centre IC"),
       col = c("black", "grey", "red"), lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
       text.col = "black", horiz = FALSE, inset=0.1)
```




```
autoplot(M_erables)
```



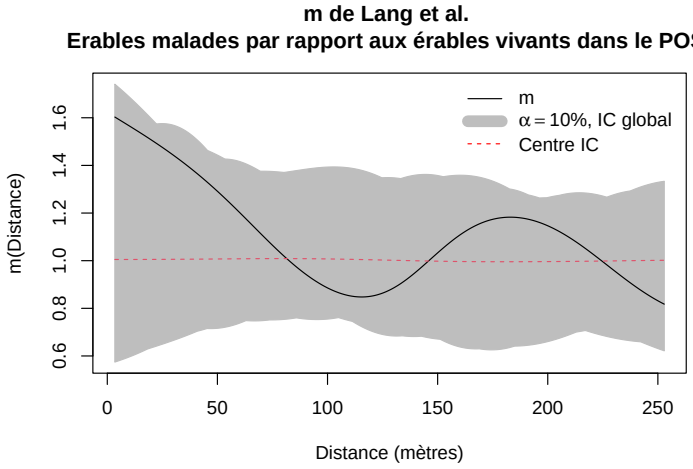

Conclusion : sign conc à 6m à 1000 sim et alpha = 10%, très légèrement sign pour alpha = 1%.

```
m_erables <- mEnvelope(BDD_V2022_A2022_Poids_POSL_erables_VPatho_wmppp,
                       NumberOfSimulations=1000,
                       Alpha=0.1, # attention : 10% ici
                       ReferenceType = "Abattu",
                       SimulationType = "RandomLocation",
                       Global=TRUE)
```

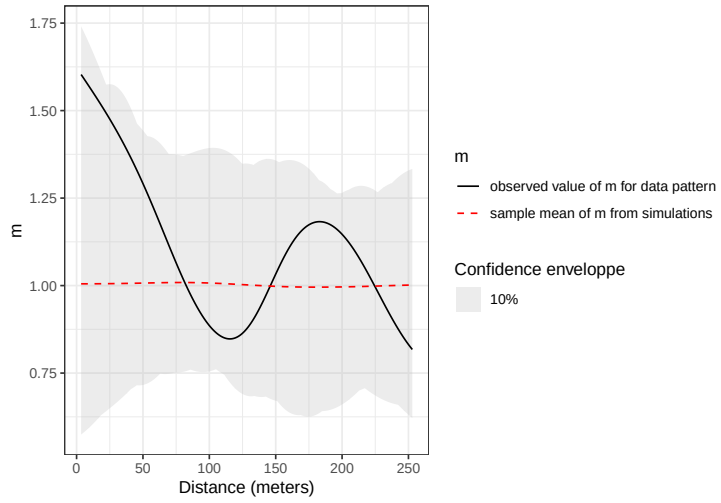
### m érables patho

```
## Generating 1000 simulations by evaluating
## expression  ...
## 1, 2, 3, ......10.........20.........30.........40..
## .......50.........60.........70.........80....
## .....90.........100.........110.........120......
## ...130.........140.........150.........160........
## .170.........180.........190.........200.........210
## .........220.........230.........240.........250..
## .......260.........270.........280.........290....
## .....300.........310.........320.........330......
## ...340.........350.........360.........370........
## .380.........390.........400.........410.........420
## .........430.........440.........450.........460..
## .......470.........480.........490.........500....
## .....510.........520.........530.........540......
## ...550.........560.........570.........580........
## .590.........600.........610.........620.........630
## .........640.........650.........660.........670..
## .......680.........690.........700.........710....
## .....720.........730.........740.........750......
## ...760.........770.........780.........790........
## .800.........810.........820.........830.........840
## .........850.........860.........870.........880..
## .......890.........900.........910.........920....
## .....930.........940.........950.........960......
## ...970.........980.........990......... 1000.
##
## Done.
```

```r
plot(m_erables,
     legend = FALSE,
     xlab="Distance (mètres)", ylab="m(Distance)",
     main="m de Lang et al. \n Erables malades par rapport aux érables vivants dans le POSL")
legend("topright",c("m", expression(alpha=='10%, IC global'), "Centre IC"),
       col = c("black", "grey", "red"), lty = c(1, 1, 8), lwd = c(1, 10, 1), bty = "n",
       text.col = "black", horiz = FALSE, inset=0.01)
```




```r
autoplot(m_erables)
```




Conclusion : NS si alpha 1%, NS le plus souvent si alpha = 10%