

Geoestatística

Bruno Mesquita dos Santos

2024-01-08

Geoestatística - Variável Dicotomica

Instalando Pacotes

```
if (!require('readxl'))install.packages("readxl");library(readxl)

## Loading required package: readxl

if (!require('fBasics'))install.packages("fBasics");library(fBasics)

## Loading required package: fBasics

if (!require('geoR'))install.packages("geoR");library(geoR)

## Loading required package: geoR

## -----
## Analysis of Geostatistical Data
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
## geoR version 1.9-3 (built on 2023-12-11) is now loaded
## -----
```

Função

```
dicotomica <- function(df,col){
  q1 = quantile(df[,col],0.25)
  q2 = quantile(df[,col],0.5)
  q3 = quantile(df[,col],0.75)
  md = mean(df[,col])

  df[paste0(col,'_q1')] = ifelse(df[,col]<=q1,1,0)
  df[paste0(col,'_q2')] = ifelse(df[,col]<=q2,1,0)
  df[paste0(col,'_md')] = ifelse(df[,col]<=md,1,0)
  df[paste0(col,'_q3')] = ifelse(df[,col]<=q3,1,0)
  df
}
```

Lendo Arquivo

```
dados <- read_excel("dados/dados1-pibic.xlsx")
dados

## # A tibble: 56 x 7
##   Latitude Longitude Altitude Umidade Argila Silte Areia
##       <dbl>     <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
## 1 300220    7803535    1291    24.6     20     28     52
## 2 300220    7803526    1274    20.1     19     28     53
## 3 300231    7803535    1287    15.4     13     29     58
## 4 300207    7803536    1290    25.6     19     24     57
## 5 300202    7803539    1289    22.9     19     25     56
## 6 300192    7803532    1285    27.2     20     30     50
## 7 300202    7803522    1277    14.7     17     28     55
## 8 300212    7803511    1277    15.6     11     19     70
## 9 300225    7803496    1278    12.6     13     28     59
## 10 300240   7803481    1277    12.5     13     28     59
## # i 46 more rows
```

Transformar para Variáveis Dicotômica

Criando uma variáveis dicotômicas para cada variável original, considerando como corte os valores de média, Q1, Q2 e Q3. A regra será se o valor original for menor ou igual ao valor de corte a dicotômica recebe 1, caso contrário recebe 0.

```
for (i in names(dados)[3:length(names(dados))]) {
  dados = dicotomica(dados,i)
}
```

Transformando os Dados em Geodados

```
geo_altitude_q1<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Altitude_q1")
geo_altitude_q2<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Altitude_q2")
geo_altitude_md<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Altitude_md")
geo_altitude_q3<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Altitude_q3")

geo_umidade_q1<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Umidade_q1")
geo_umidade_q2<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Umidade_q2")
geo_umidade_md<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Umidade_md")
geo_umidade_q3<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                               data.col="Umidade_q3")
```

```

geo_argila_q1<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Argila_q1")
geo_argila_q2<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Argila_q2")
geo_argila_md<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Argila_md")
geo_argila_q3<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Argila_q3")

geo_silte_q1<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Silte_q1")
geo_silte_q2<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Silte_q2")
geo_silte_md<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Silte_md")
geo_silte_q3<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Silte_q3")

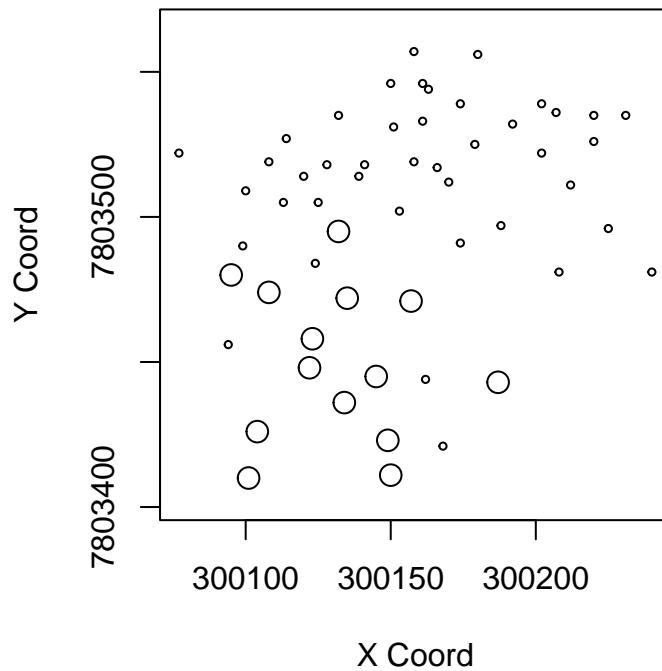
geo_areia_q1<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Areia_q1")
geo_areia_q2<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Areia_q2")
geo_areia_md<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Areia_md")
geo_areia_q3<-as.geodata(dados,coords.col=c("Latitude","Longitude"),
                           data.col="Areia_q3")

```

Gráficos Espaciais

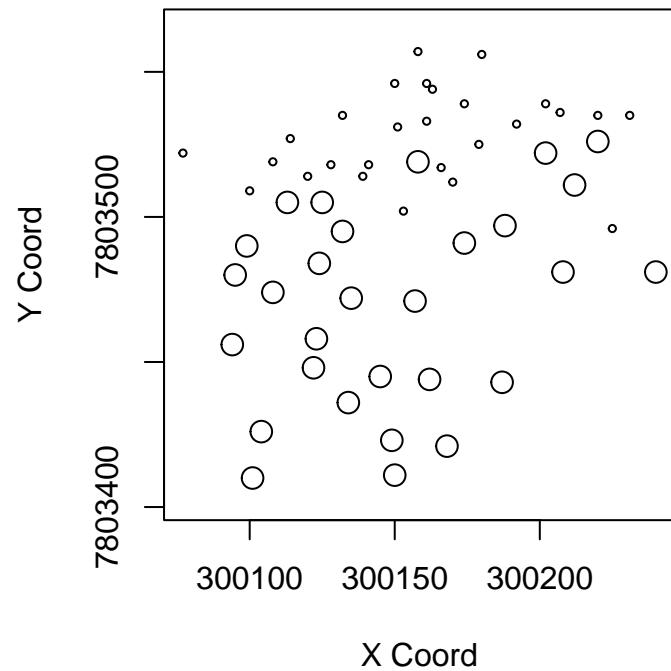
```
points(geo_altitude_q1, main="Semivariograma altitude da dicotomica para Q1")
```

Semivariograma altitude da dicotomica para Q1



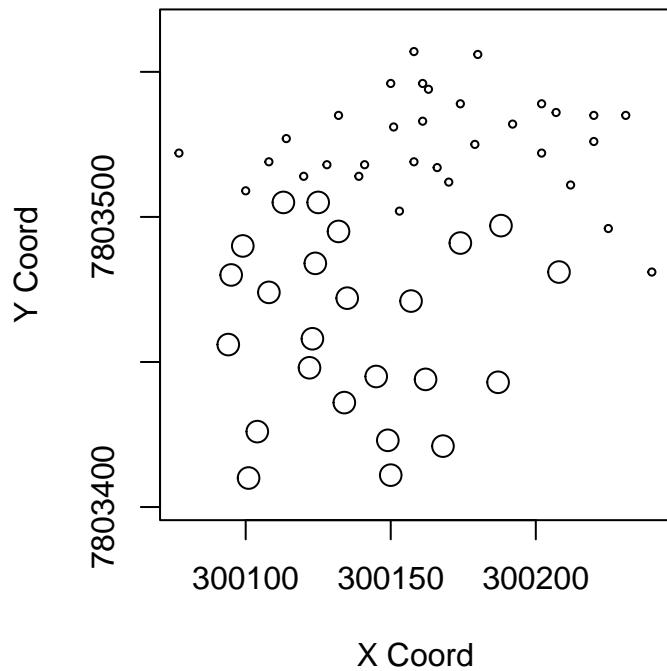
```
points(geo_altitude_q2, main="Semivariograma altitude da dicotomica para Q2")
```

Semivariograma altitude da dicotomica para Q2



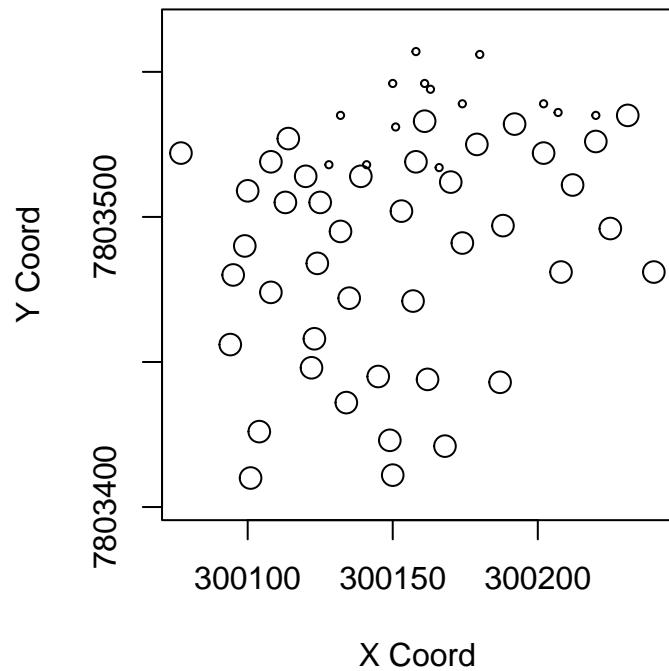
```
points(geo_altitude_md, main="Semivariograma altitude da dicotomica para Médio")
```

Semivariograma altitude da dicotomica para Médio



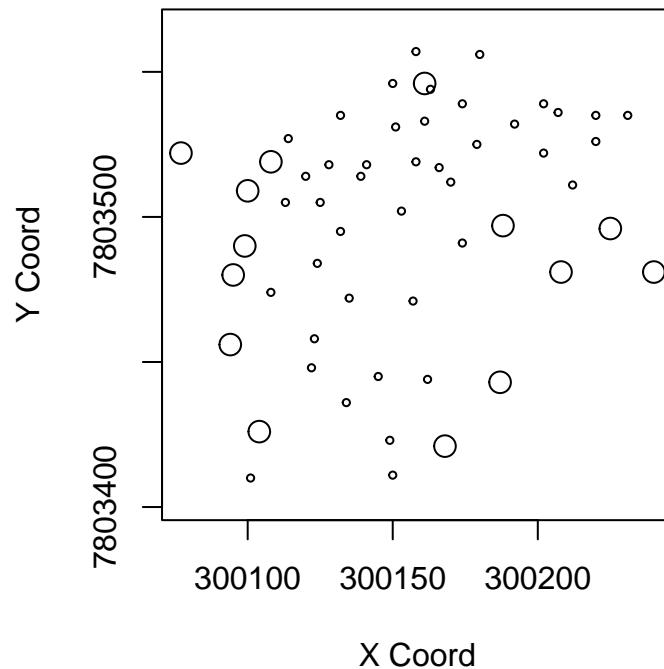
```
points(geo_altitude_q3, main="Semivariograma altitude da dicotomica para Q3")
```

Semivariograma altitude da dicotomica para Q3



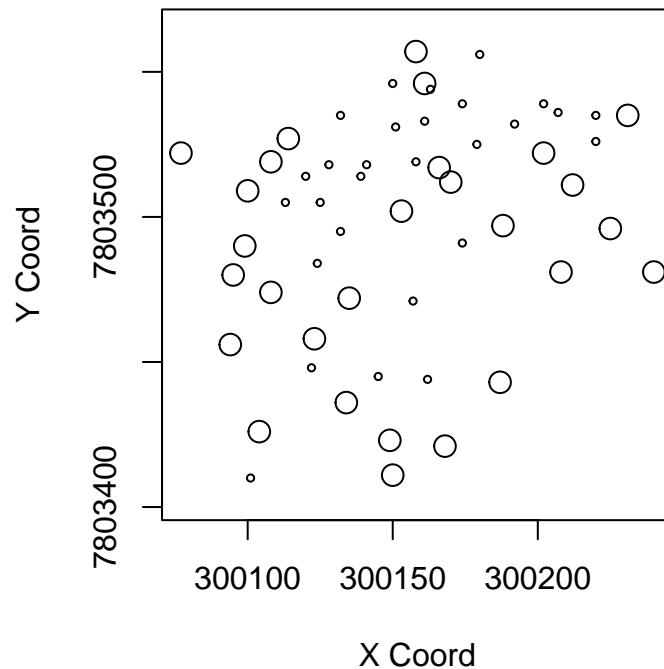
```
points(geo_umidade_q1, main="Semivariograma umidade da dicotomica para Q1")
```

Semivariograma umidade da dicotomica para Q1



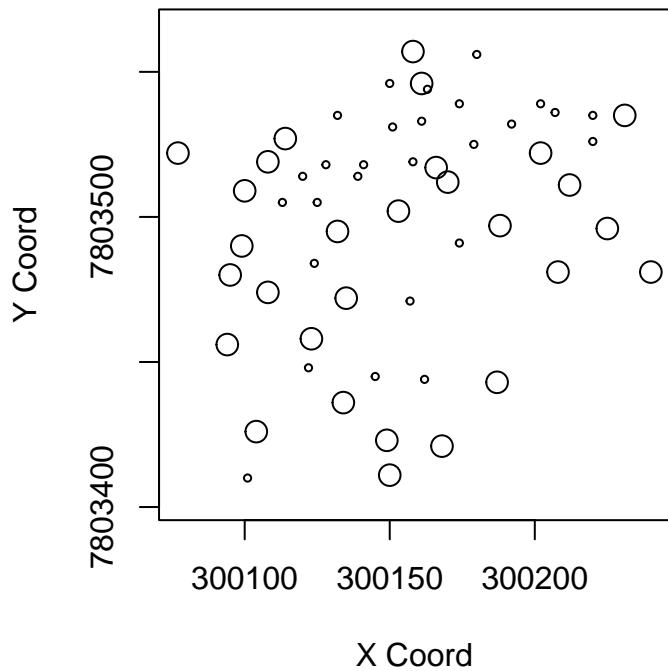
```
points(geo_umidade_q2, main="Semivariograma umidade da dicotomica para Q2")
```

Semivariograma umidade da dicotomica para Q2



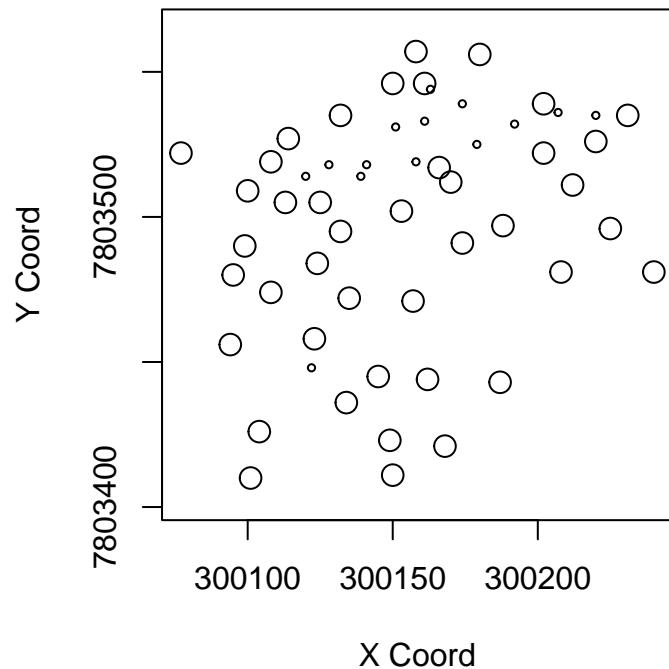
```
points(geo_umidade_md, main="Semivariograma umidade da dicotomica para Médio")
```

Semivariograma umidade da dicotomica para Médio



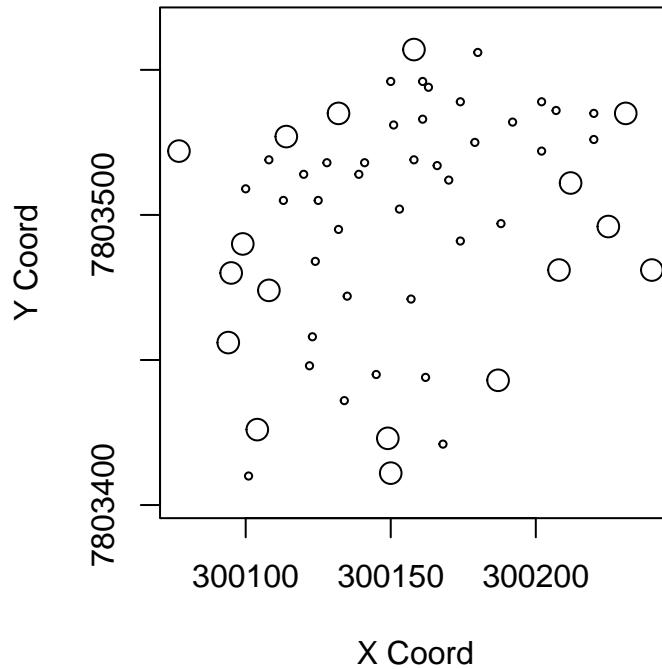
```
points(geo_umidade_q3, main="Semivariograma umidade da dicotomica para Q3")
```

Semivariograma umidade da dicotomica para Q3



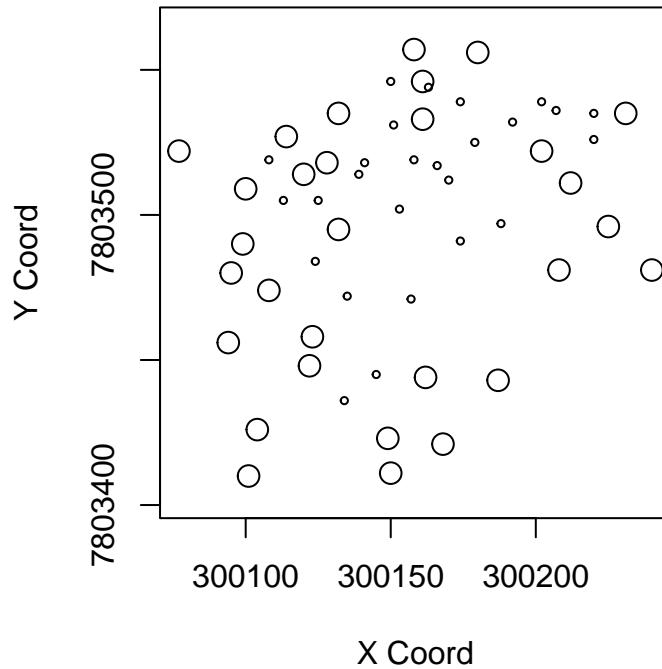
```
points(geo_argila_q1, main="Semivariograma argila da dicotomica para Q1")
```

Semivariograma argila da dicotomica para Q1



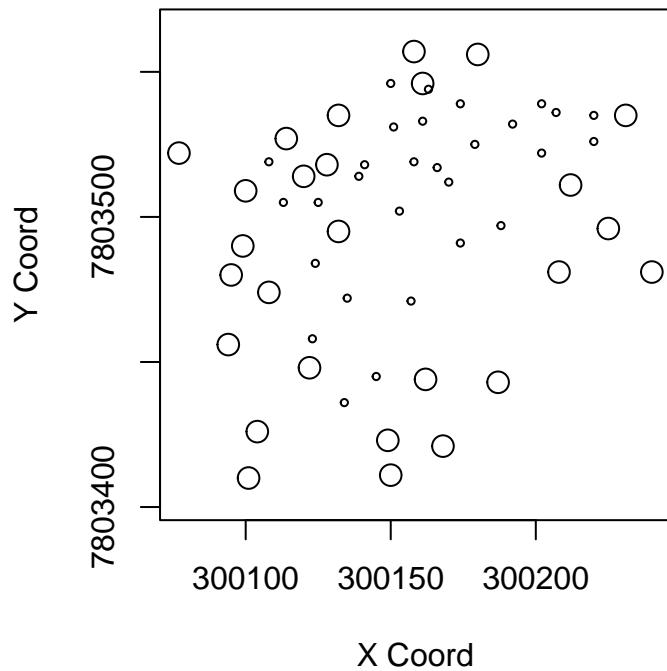
```
points(geo_argila_q2, main="Semivariograma argila da dicotomica para Q2")
```

Semivariograma argila da dicotomica para Q2



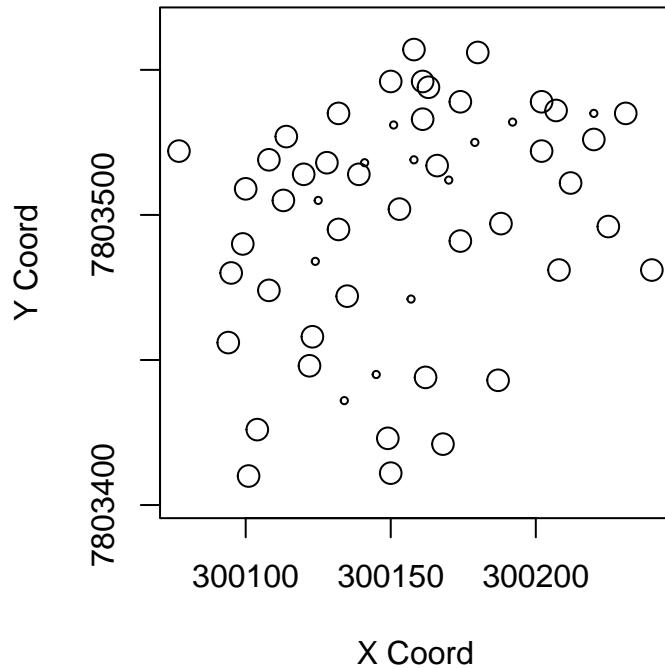
```
points(geo_argila_md, main="Semivariograma argila da dicotomica para Médio")
```

Semivariograma argila da dicotomica para Médio



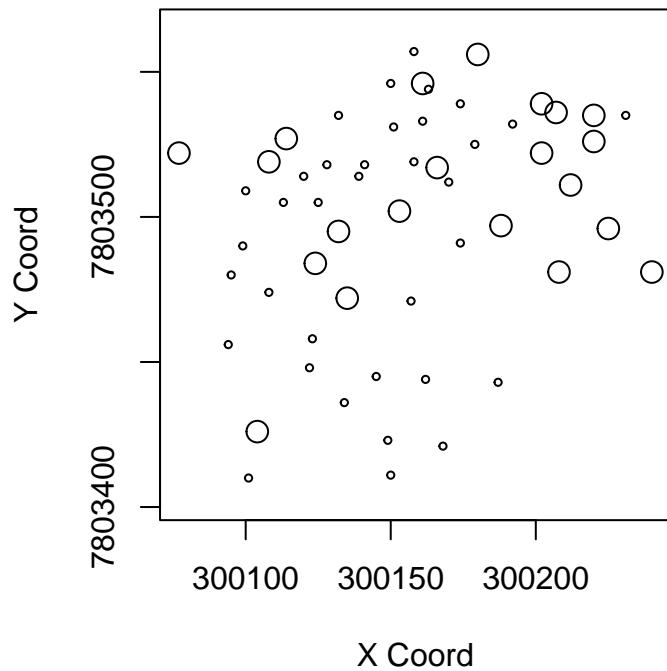
```
points(geo_argila_q3, main="Semivariograma argila da dicotomica para Q3")
```

Semivariograma argila da dicotomica para Q3



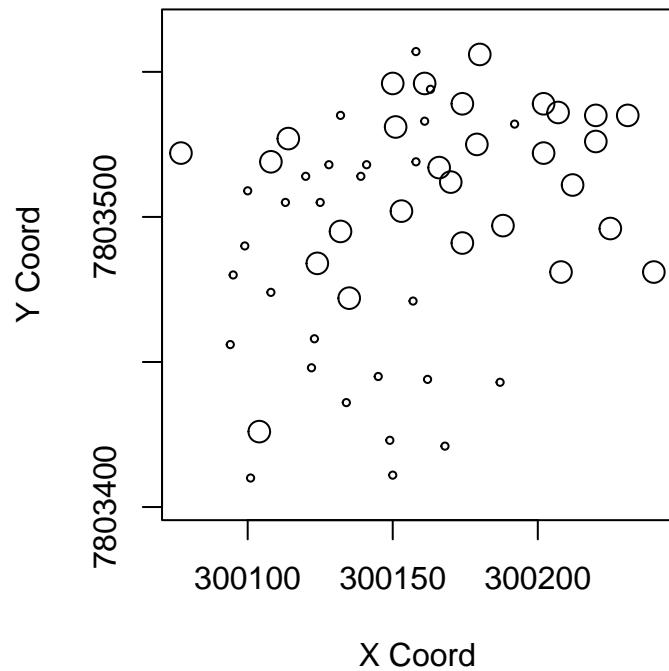
```
points(geo_silte_q1, main="Semivariograma silte da dicotomica para Q1")
```

Semivariograma silte da dicotomica para Q1



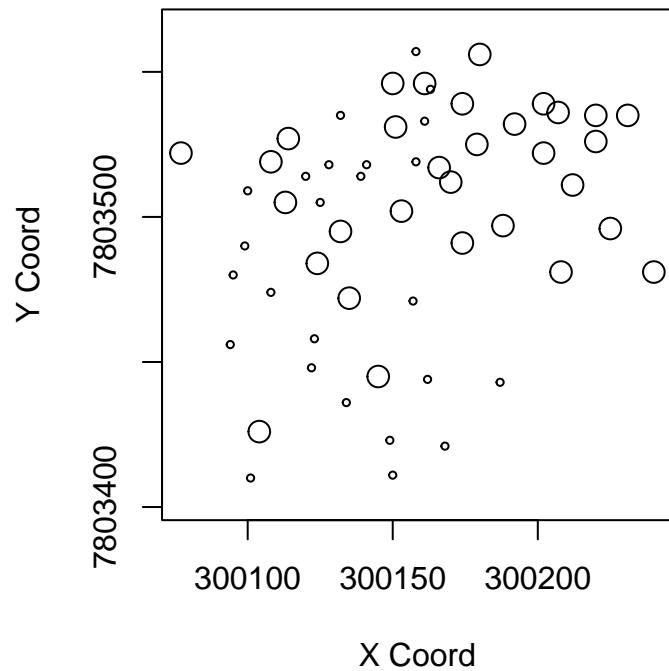
```
points(geo_silte_q2, main="Semivariograma silte da dicotomica para Q2")
```

Semivariograma silte da dicotomica para Q2



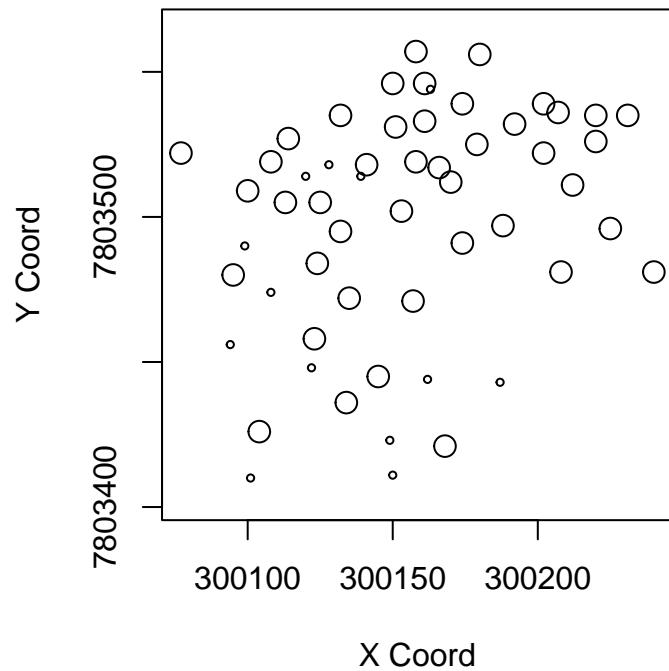
```
points(geo_silte_md, main="Semivariograma silte da dicotomica para Médio")
```

Semivariograma silte da dicotomica para Médio



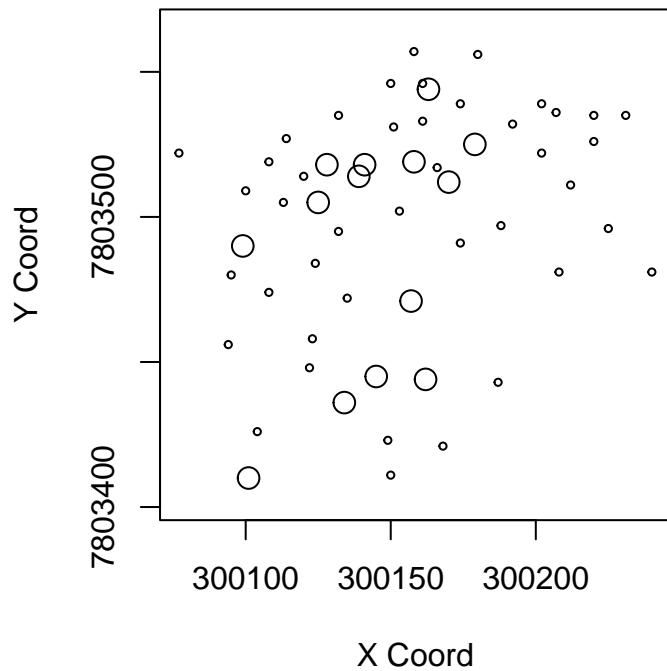
```
points(geo_silte_q3, main="Semivariograma silte da dicotomica para Q3")
```

Semivariograma silte da dicotomica para Q3



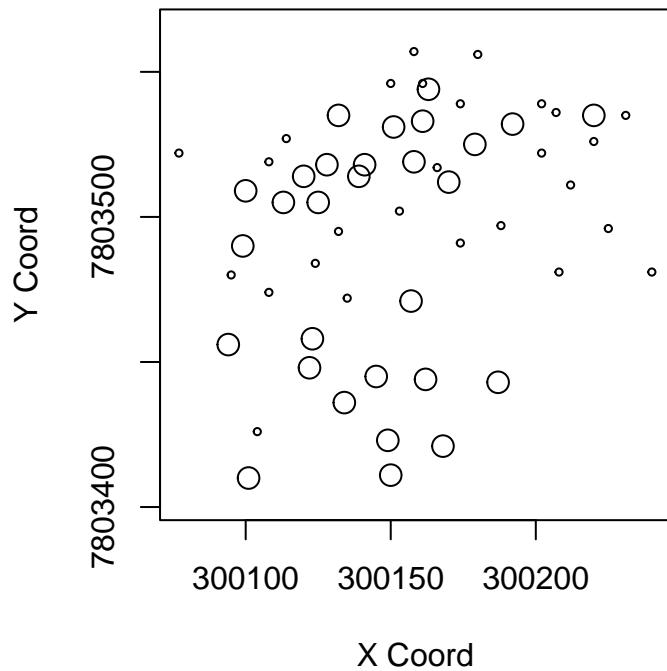
```
points(geo_areia_q1, main="Semivariograma areia da dicotomica para Q1")
```

Semivariograma areia da dicotomica para Q1



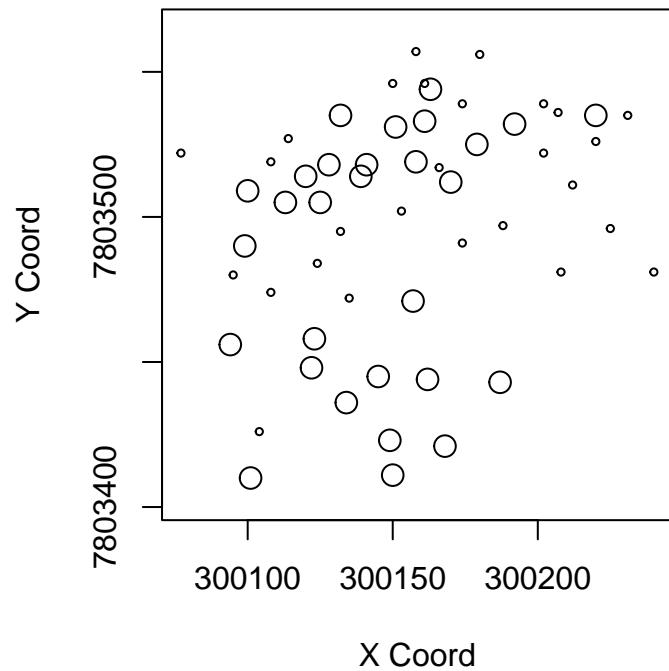
```
points(geo_areia_q2, main="Semivariograma areia da dicotomica para Q2")
```

Semivariograma areia da dicotomica para Q2



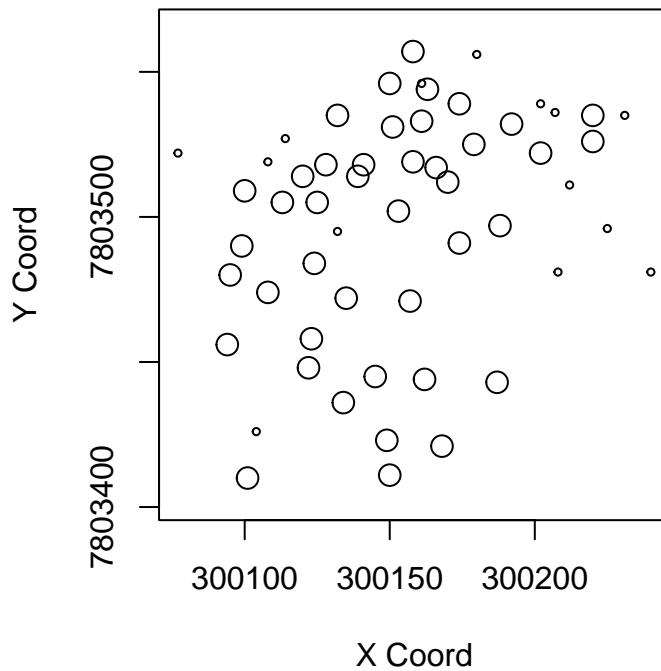
```
points(geo_areia_md, main="Semivariograma areia da dicotomica para Médio")
```

Semivariograma areia da dicotomica para Médio



```
points(geo_areia_q3, main="Semivariograma areia da dicotomica para Q3")
```

Semivariograma areia da dicotomica para Q3



Ajustando o Semivariograma

```
semi_altitude_q1 <- variog(geo_altitude_q1, max.dist=50)

## variog: computing omnidirectional variogram

plot(semi_altitude_q1, main="Semivariograma altitude da dicotomica para Q1")
ajust_altitude_q1 <- variofit(semi_altitude_q1, max.dist=50,wei="equal")

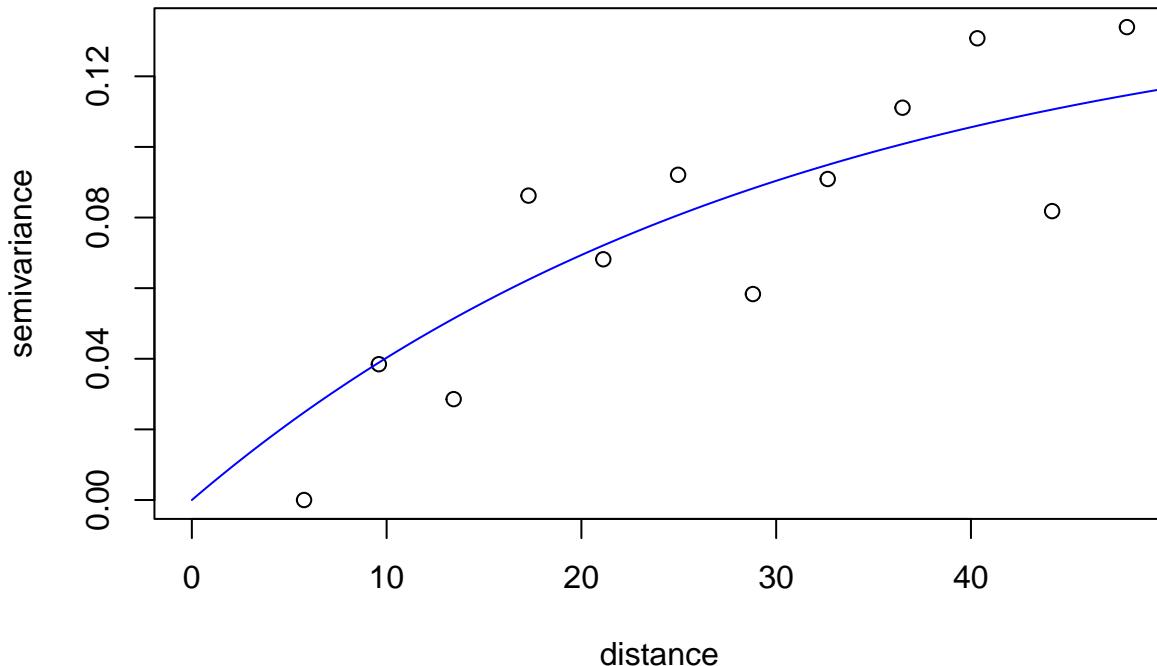
## variofit: covariance model used is matern
## variofit: weights used: equal
## variofit: minimisation function used: optim

## Warning in variofit(semi_altitude_q1, max.dist = 50, wei = "equal"): initial
## values not provided - running the default search

## variofit: searching for best initial value ... selected values:
##           sigmasq phi      tausq kappa
## initial.value "0.13" "30.73" "0"     "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 0.00517044702182927
```

```
lines(ajust_altitude_q1, col="blue")
```

Semivariograma altitude da dicotomica para Q1



```
semi_altitude_q2 <- variog(geo_altitude_q2, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_altitude_q2, main="Semivariograma altitude da dicotomica para Q2")
ajust_altitude_q2 <- variofit(semi_altitude_q2, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_altitude_q2, max.dist = 50, wei = "equal"): initial
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq phi      tausq kappa
```

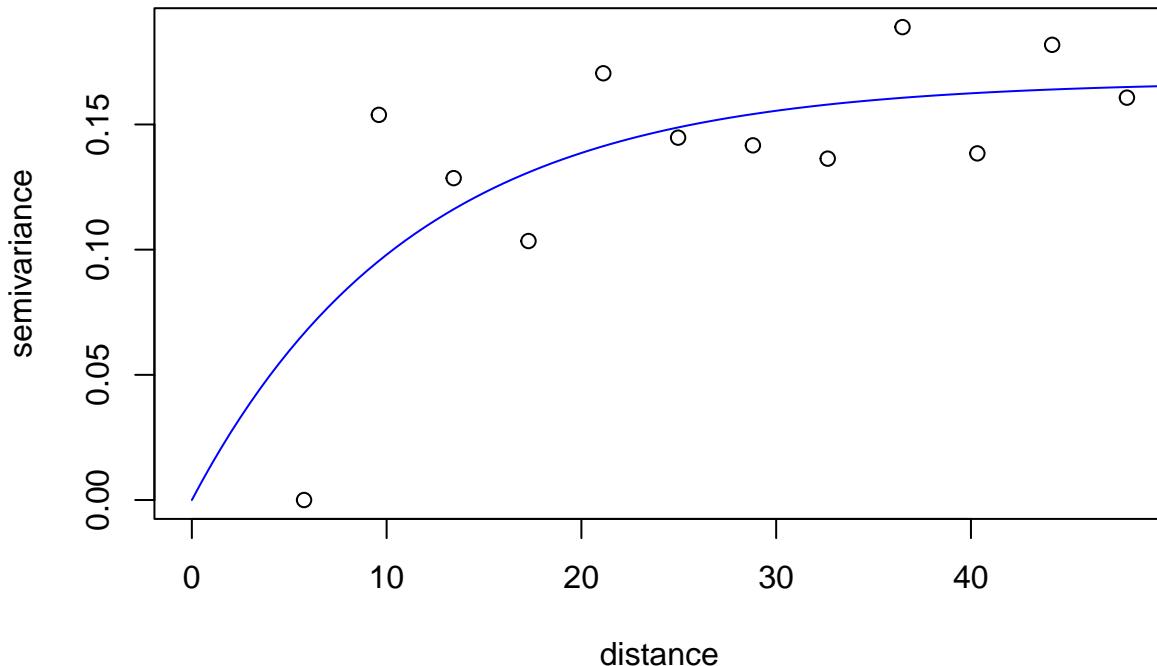
```
## initial.value "0.19" "15.36" "0"    "0.5"
```

```
## status       "est"   "est"   "est"  "fix"
```

```
## loss value: 0.0129804824125576
```

```
lines(ajust_altitude_q2, col="blue")
```

Semivariograma altitude da dicotomica para Q2



```
semi_altitude_md <- variog(geo_altitude_md, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_altitude_md, main="Semivariograma altitude da dicotomica para Médio")  
ajust_altitude_md <- variofit(semi_altitude_md, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_altitude_md, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq phi      tausq kappa
```

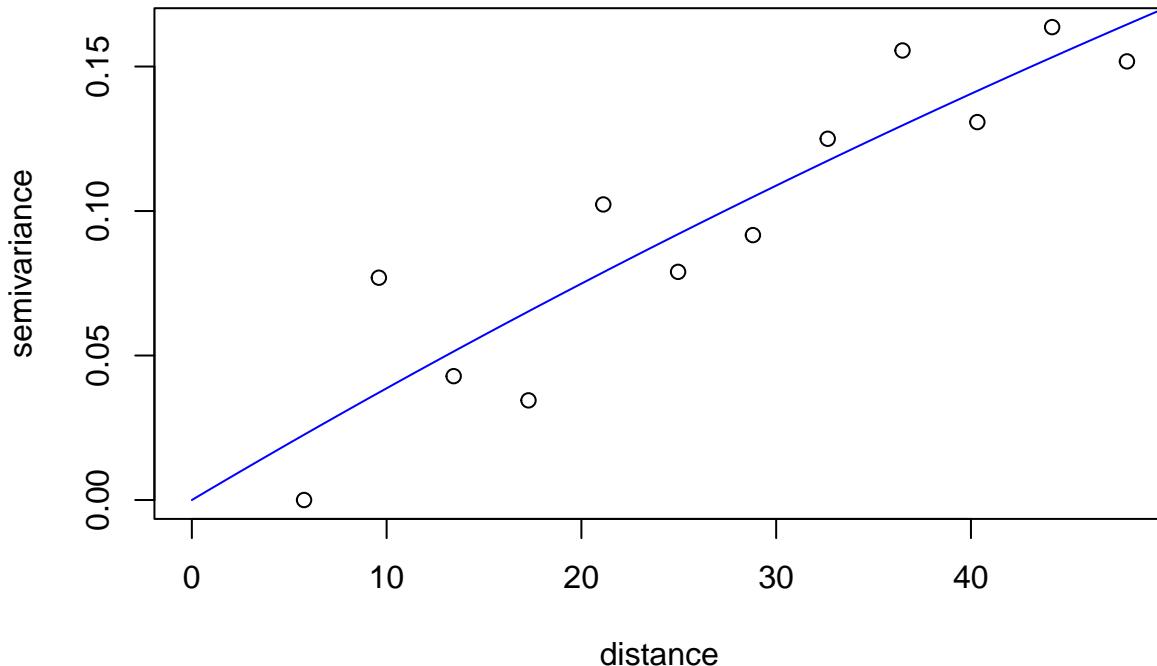
```
## initial.value "0.16" "23.04" "0"    "0.5"
```

```
## status       "est"   "est"   "est" "fix"
```

```
## loss value: 0.00814359736507235
```

```
lines(ajust_altitude_md, col="blue")
```

Semivariograma altitude da dicotomica para Médio



```
semi_altitude_q3 <- variog(geo_altitude_q3, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_altitude_q3, main="Semivariograma altitude da dicotomica para Q3")
ajust_altitude_q3 <- variofit(semi_altitude_q1, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_altitude_q1, max.dist = 50, wei = "equal"): initial
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq phi      tausq kappa
```

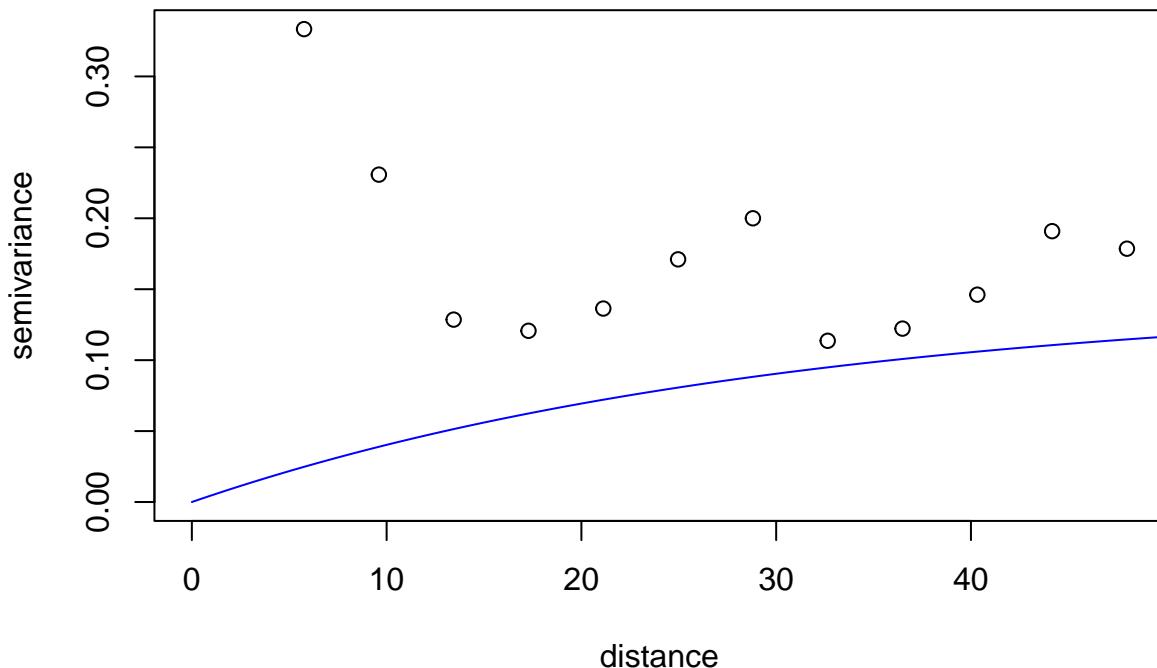
```
## initial.value "0.13" "30.73" "0"    "0.5"
```

```
## status       "est"   "est"   "est"  "fix"
```

```
## loss value: 0.00517044702182927
```

```
lines(ajust_altitude_q3, col="blue")
```

Semivariograma altitude da dicotomica para Q3



```
semi_umidade_q1 <- variog(geo_umidade_q1, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_umidade_q1, main="Semivariograma umidade da dicotomica para Q1")  
ajust_umidade_q1 <- variofit(semi_umidade_q1, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_umidade_q1, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi     tausq   kappa
```

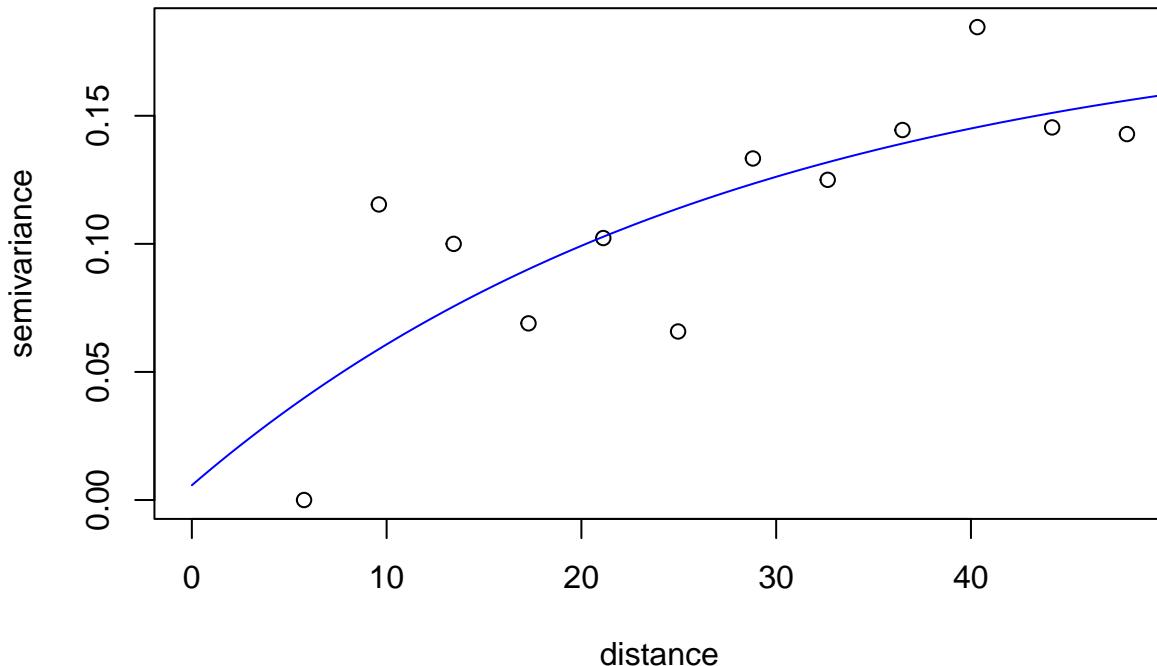
```
## initial.value "0.18" "38.41" "0.02" "0.5"
```

```
## status       "est"    "est"    "est"    "fix"
```

```
## loss value: 0.0104544311390305
```

```
lines(ajust_umidade_q1, col="blue")
```

Semivariograma umidade da dicotomica para Q1



```
semi_umidade_q2 <- variog(geo_umidade_q2, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_umidade_q2, main="Semivariograma umidade da dicotomica para Q2")  
ajust_umidade_q2 <- variofit(semi_umidade_q2, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_umidade_q2, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

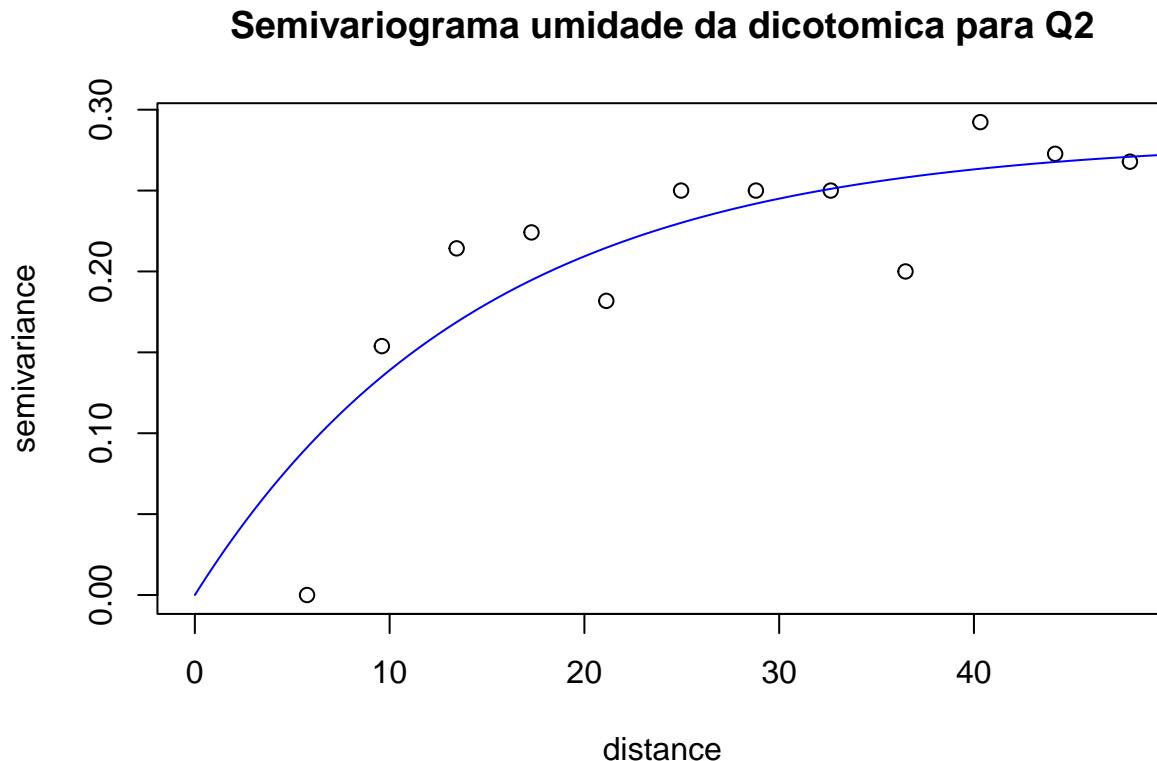
```
##           sigmasq phi      tausq kappa
```

```
## initial.value "0.29"  "15.36" "0"    "0.5"
```

```
## status        "est"   "est"   "est"  "fix"
```

```
## loss value: 0.0177798912432688
```

```
lines(ajust_umidade_q2, col="blue")
```



```
semi_umidade_md <- variog(geo_umidade_md, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_umidade_md, main="Semivariograma umidade da dicotomica para Médio")  
ajust_umidade_md <- variofit(semi_umidade_md, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_umidade_md, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

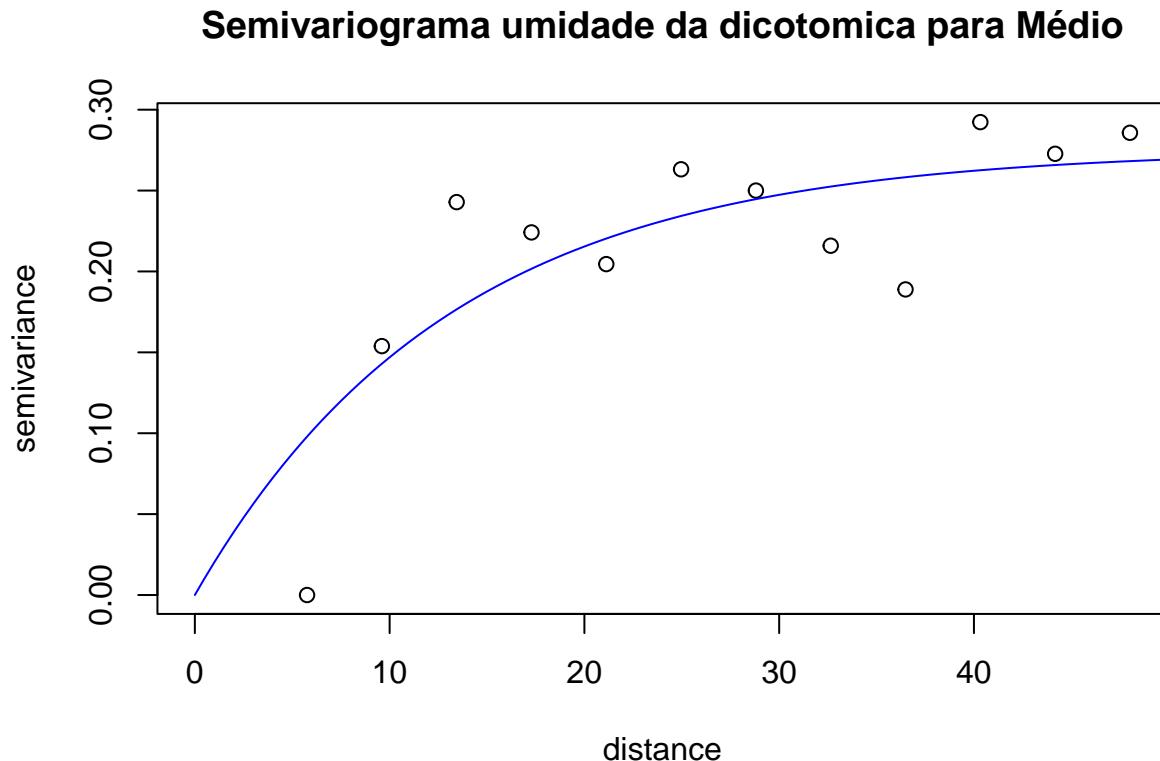
```
##           sigmasq phi      tausq kappa
```

```
## initial.value "0.29" "15.36" "0"     "0.5"
```

```
## status        "est"   "est"    "est"  "fix"
```

```
## loss value: 0.0236703158917658
```

```
lines(ajust_umidade_md, col="blue")
```



```
semi_umidade_q3 <- variog(geo_umidade_q3, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_umidade_q3, main="Semivariograma umidade da dicotomica para Q3")  
ajust_umidade_q3 <- variofit(semi_umidade_q3, max.dist=50, wei="equal")
```

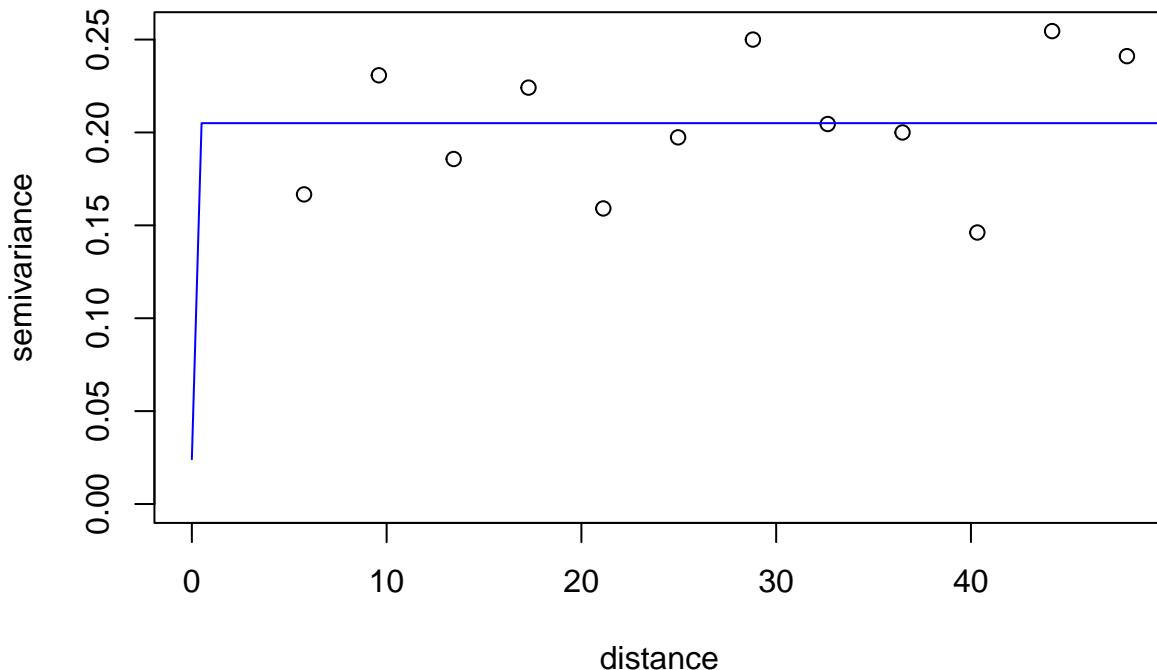
```
## variofit: covariance model used is matern  
## variofit: weights used: equal  
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_umidade_q3, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:  
##           sigmasq   phi    tausq   kappa  
## initial.value "0.19" "0"    "0.03" "0.5"  
## status        "est"   "est"   "est"   "fix"  
## loss value: 0.0158546996460259
```

```
lines(ajust_umidade_q3, col="blue")
```

Semivariograma umidade da dicotomica para Q3



```
semi_argila_q1 <- variolog(geo_argila_q1, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_argila_q1, main="Semivariograma argila da dicotomica para Q1")  
ajust_argila_q1 <- variofit(semi_argila_q1, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_argila_q1, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq phi      tausq kappa
```

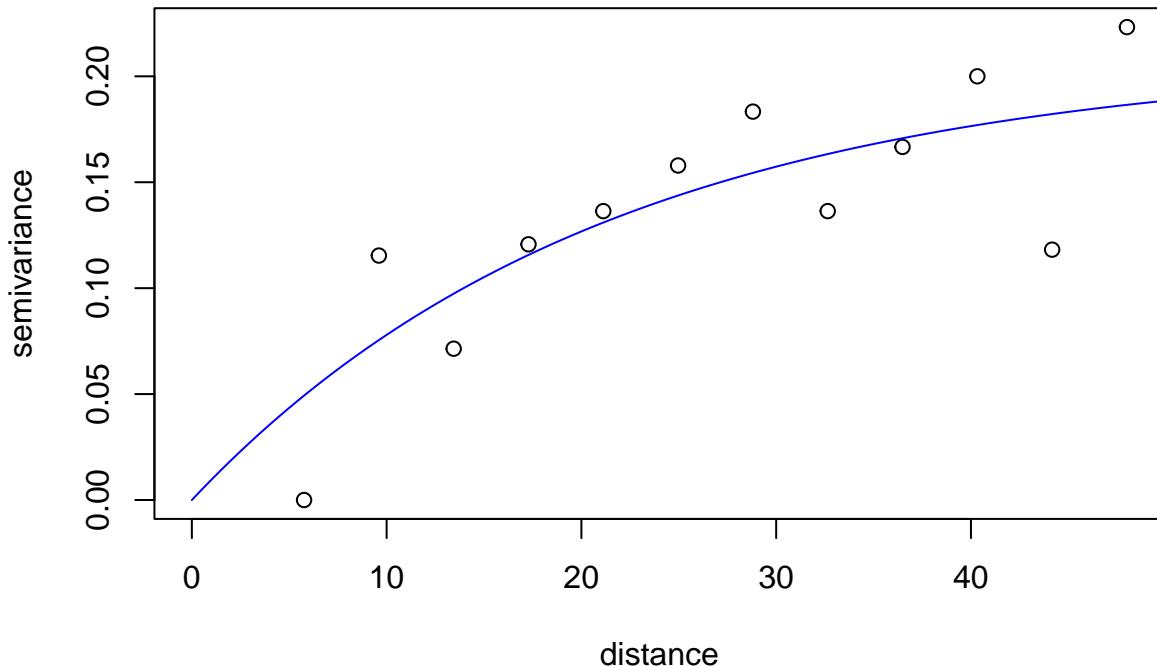
```
## initial.value "0.22" "23.04" "0"     "0.5"
```

```
## status       "est"    "est"    "est"  "fix"
```

```
## loss value: 0.0128366452642855
```

```
lines(ajust_argila_q1, col="blue")
```

Semivariograma argila da dicotomica para Q1



```
semi_argila_q2 <- variog(geo_argila_q2, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_argila_q2, main="Semivariograma argila da dicotomica para Q2")  
ajust_argila_q2 <- variofit(semi_argila_q2, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_argila_q2, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi    tausq   kappa
```

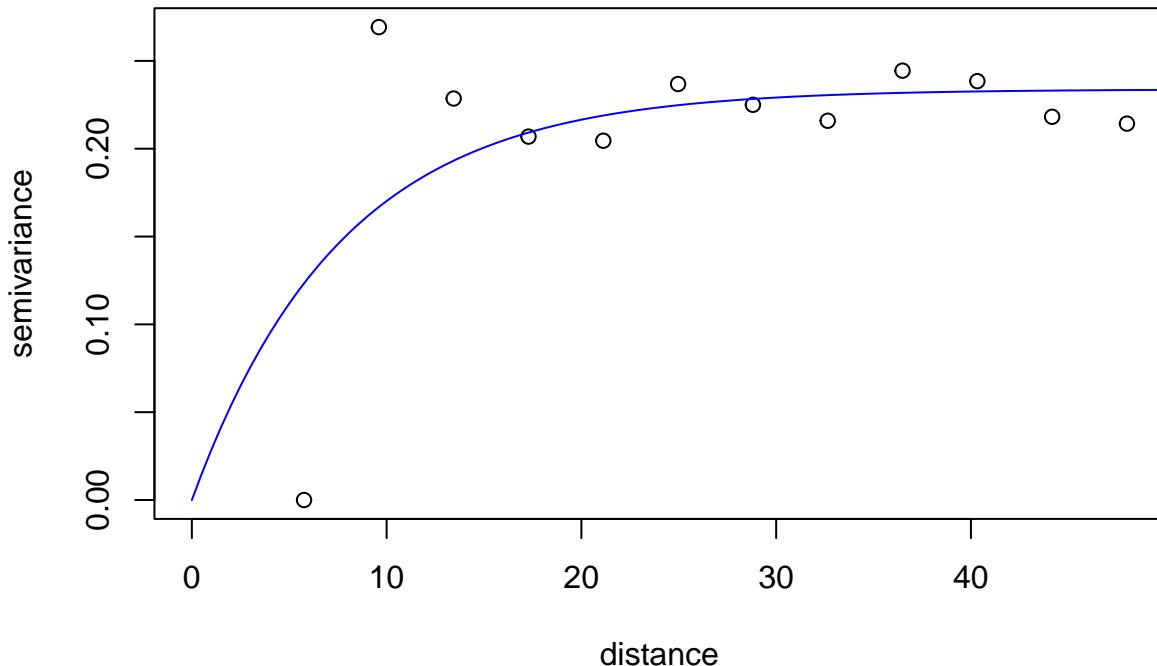
```
## initial.value "0.2"   "7.68" "0.03" "0.5"
```

```
## status       "est"   "est"   "est"   "fix"
```

```
## loss value: 0.0298738854826642
```

```
lines(ajust_argila_q2, col="blue")
```

Semivariograma argila da dicotomica para Q2



```
semi_argila_md <- variog(geo_argila_q1, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_argila_md, main="Semivariograma argila da dicotomica para Médio")  
ajust_argila_md <- variofit(semi_argila_md, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_argila_md, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi     tausq   kappa
```

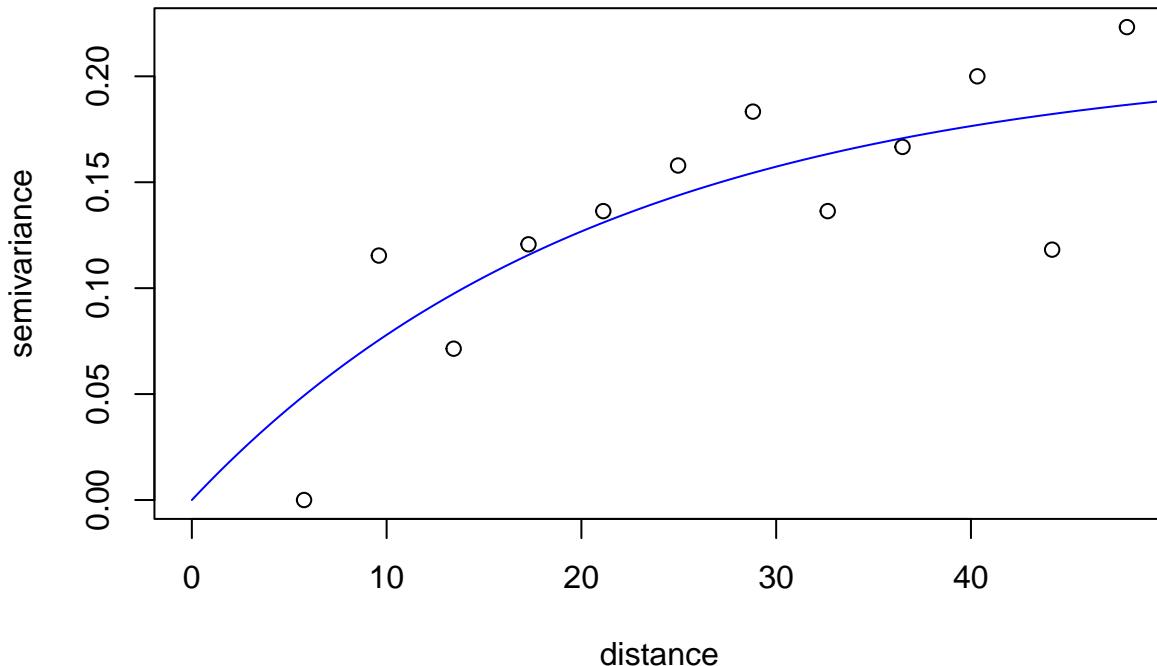
```
## initial.value "0.22" "23.04" "0"    "0.5"
```

```
## status       "est"    "est"    "est"  "fix"
```

```
## loss value: 0.0128366452642855
```

```
lines(ajust_argila_md, col="blue")
```

Semivariograma argila da dicotomica para Médio



```
semi_argila_q3 <- variolog(geo_argila_q3, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_argila_q3, main="Semivariograma argila da dicotomica para Q3")  
ajust_argila_q3 <- variofit(semi_argila_q3, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_argila_q3, max.dist = 50, wei = "equal"): initial  
## values not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi    tausq   kappa
```

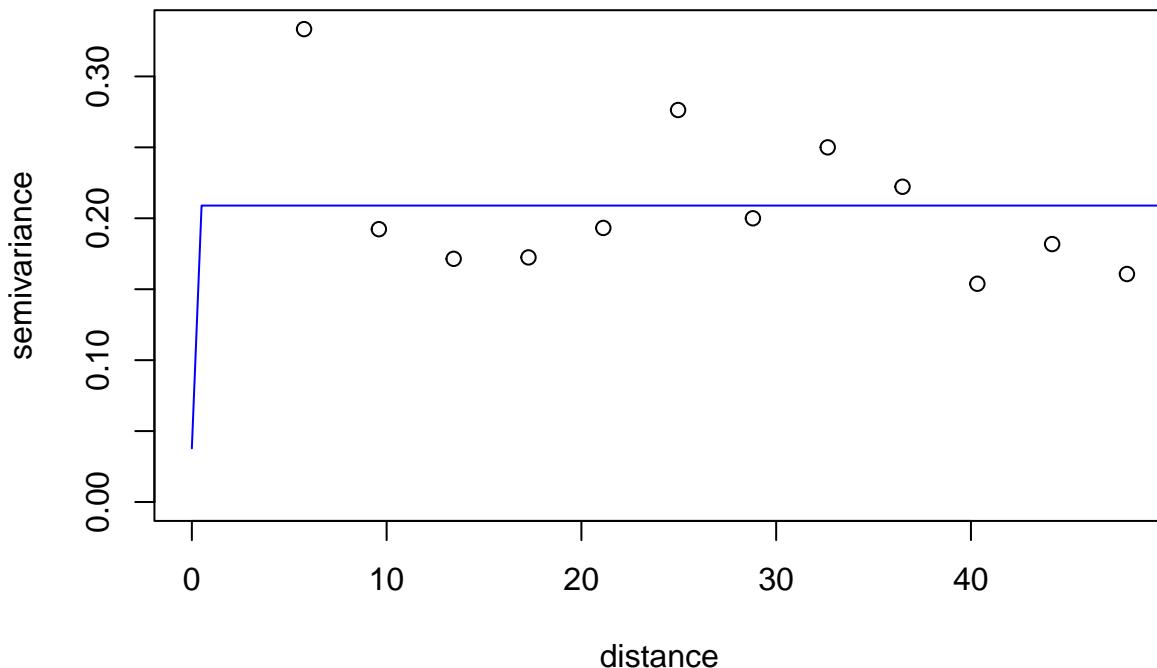
```
## initial.value "0.17" "0" "0.03" "0.5"
```

```
## status       "est" "est" "est" "fix"
```

```
## loss value: 0.0322828125778822
```

```
lines(ajust_argila_q3, col="blue")
```

Semivariograma argila da dicotomica para Q3



```
semi_silte_q1 <- variog(geo_silte_q1, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_silte_q1, main="Semivariograma silte da dicotomica para Q1")  
ajust_silte_q1 <- variofit(semi_silte_q1, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_silte_q1, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

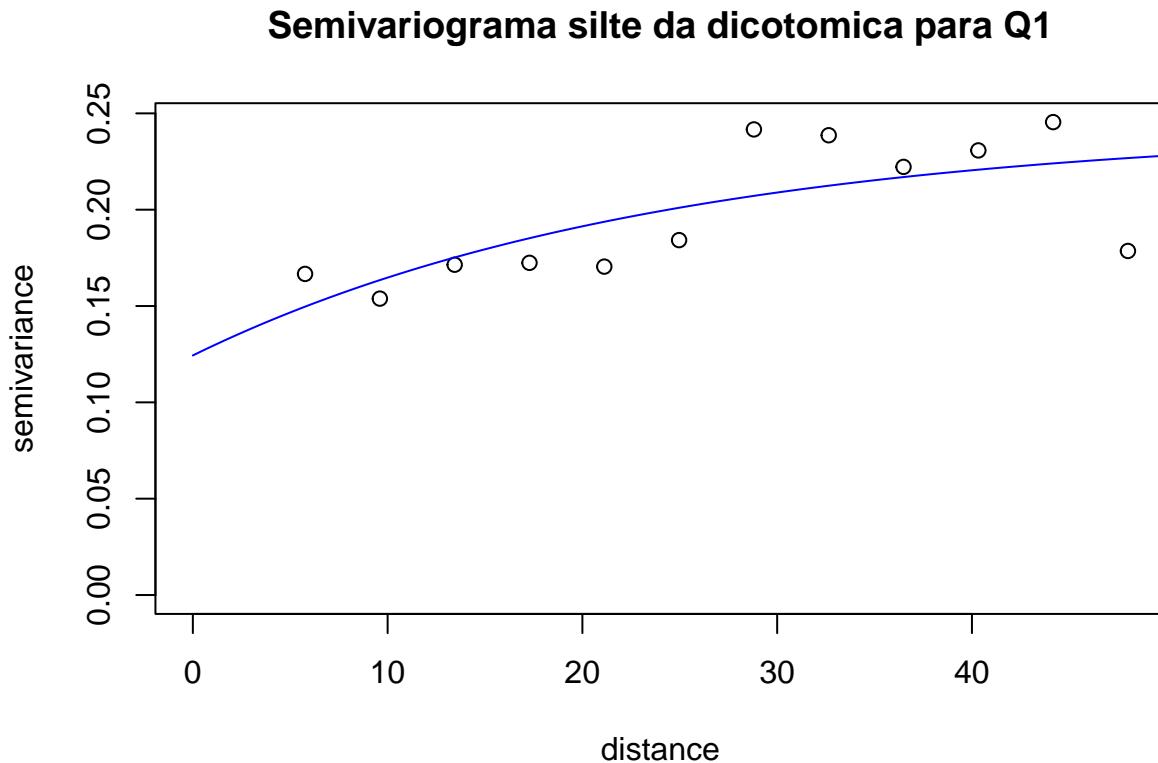
```
##           sigmasq   phi     tausq   kappa
```

```
## initial.value "0.12" "23.04" "0.12" "0.5"
```

```
## status       "est"    "est"    "est"    "fix"
```

```
## loss value: 0.00626222736000728
```

```
lines(ajust_silte_q1, col="blue")
```



```
semi_silte_q2 <- variog(geo_silte_q2, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_silte_q2, main="Semivariograma silte da dicotomica para Q2")  
ajust_silte_q2 <- variofit(semi_silte_q2, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_silte_q2, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi     tausq   kappa
```

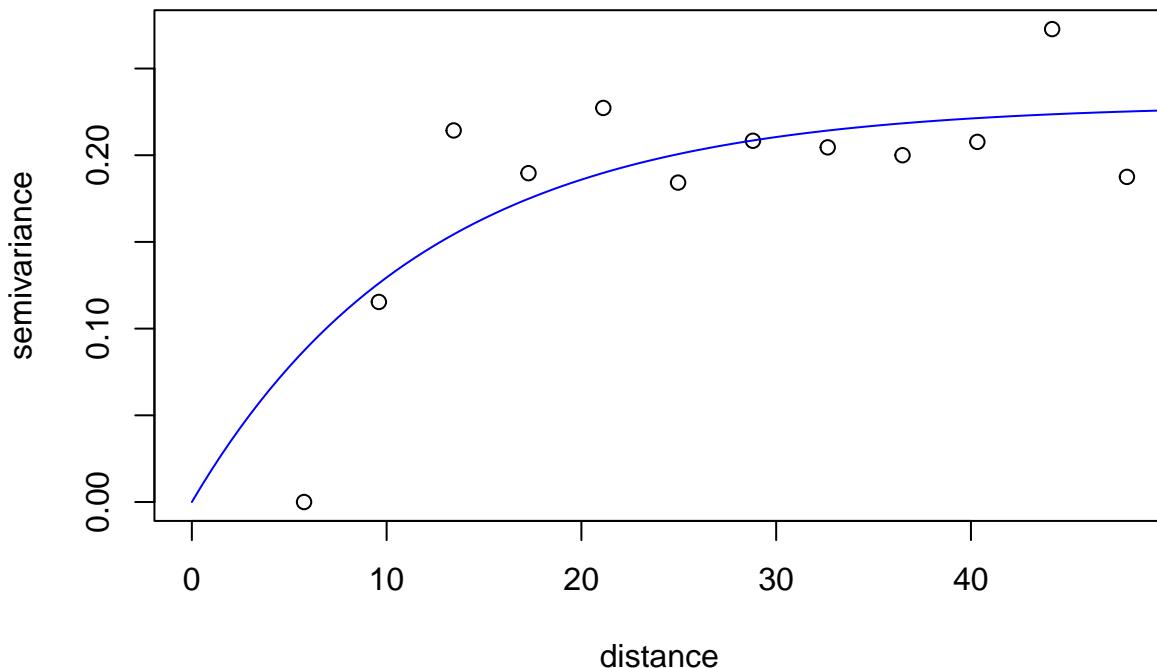
```
## initial.value "0.2"    "15.36" "0.03" "0.5"
```

```
## status        "est"    "est"    "est"    "fix"
```

```
## loss value: 0.0201466586052663
```

```
lines(ajust_silte_q2, col="blue")
```

Semivariograma silte da dicotomica para Q2



```
semi_silte_md <- variog(geo_silte_md, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_silte_md, main="Semivariograma silte da dicotomica para Médio")  
ajust_silte_md <- variofit(semi_silte_md, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_silte_md, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi    tausq   kappa
```

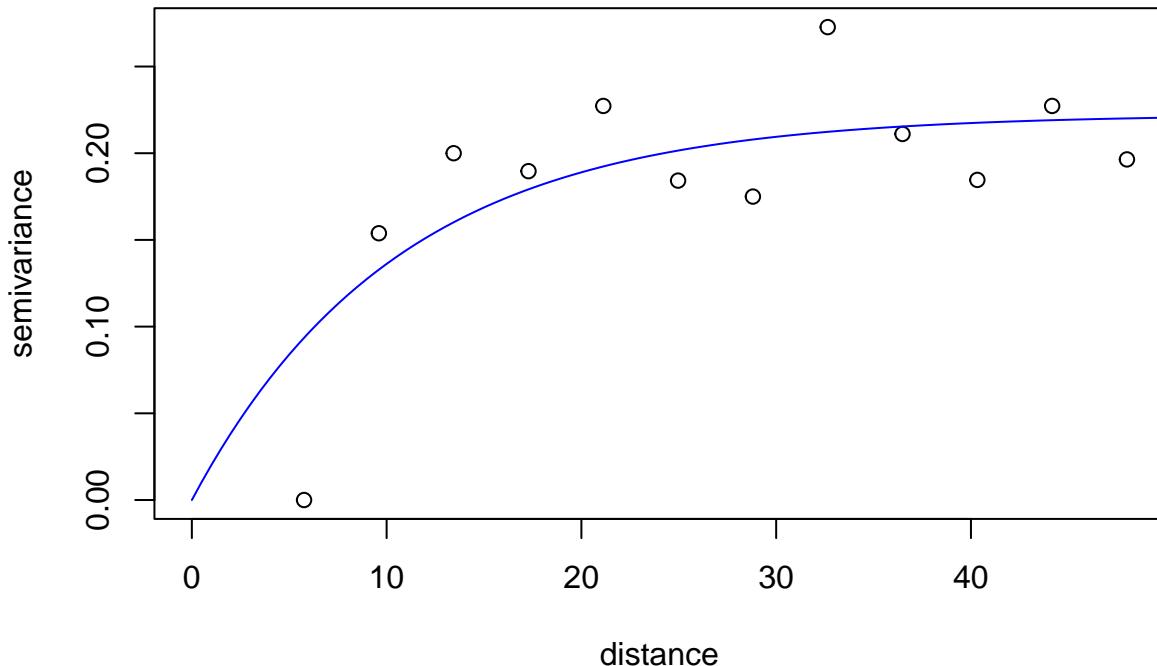
```
## initial.value "0.2"   "7.68" "0"     "0.5"
```

```
## status        "est"    "est"   "est"   "fix"
```

```
## loss value: 0.0208496332913277
```

```
lines(ajust_silte_md, col="blue")
```

Semivariograma silte da dicotomica para Médio



```
semi_silte_q3 <- variog(geo_silte_q3, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

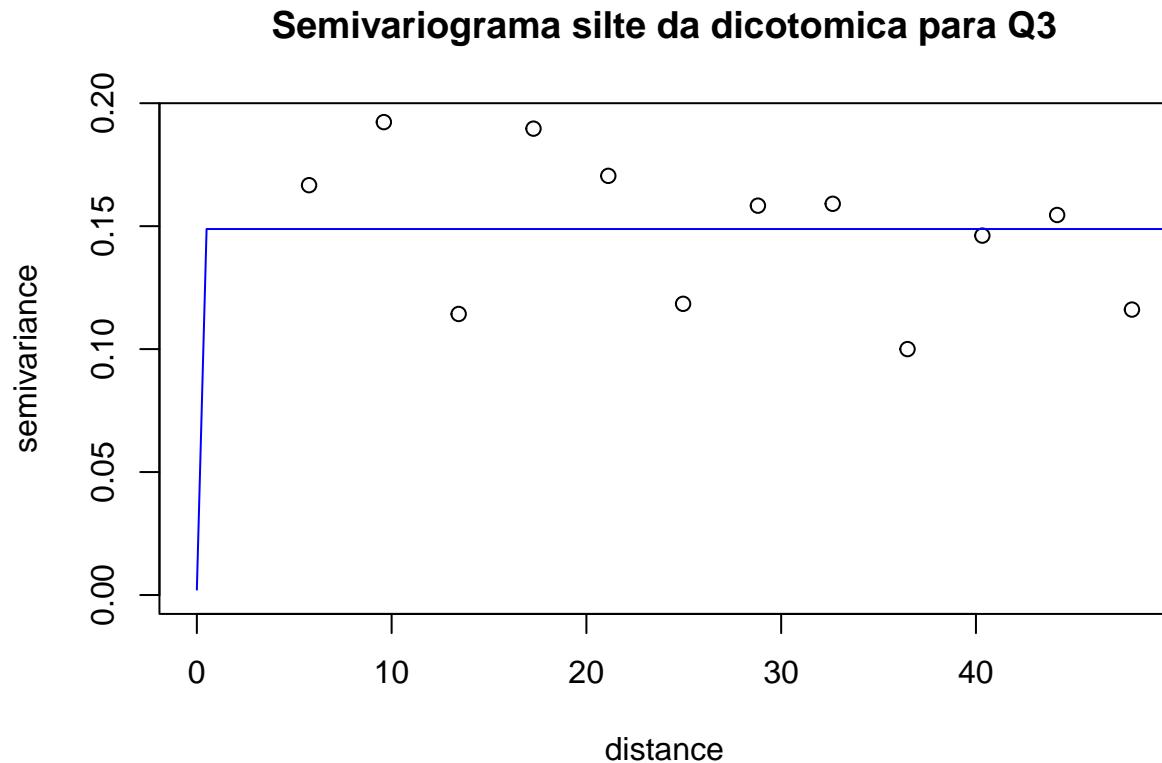
```
plot(semi_silte_q3, main="Semivariograma silte da dicotomica para Q3")  
ajust_silte_q3 <- variofit(semi_silte_q3, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern  
## variofit: weights used: equal  
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_silte_q3, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:  
##           sigmasq   phi    tausq   kappa  
## initial.value "0.14" "0"     "0"     "0.5"  
## status        "est"   "est"   "est"   "fix"  
## loss value: 0.0104077770988972
```

```
lines(ajust_silte_q3, col="blue")
```



```
semi_areia_q1 <- variog(geo_areia_q1, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_areia_q1, main="Semivariograma areia da dicotomica para Q1")  
ajust_areia_q1 <- variofit(semi_areia_q1, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_areia_q1, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

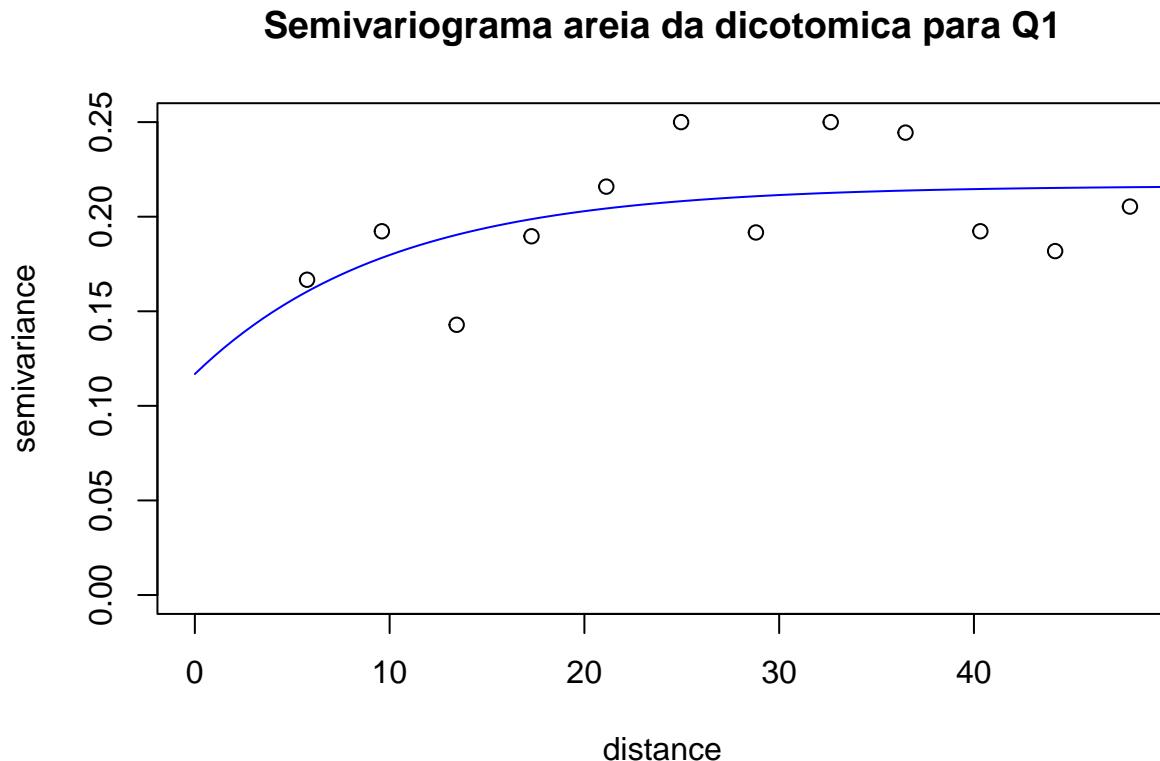
```
##           sigmasq   phi     tausq   kappa
```

```
## initial.value "0.12" "23.04" "0.12" "0.5"
```

```
## status       "est"    "est"    "est"    "fix"
```

```
## loss value: 0.0106986260181029
```

```
lines(ajust_areia_q1, col="blue")
```



```
semi_areia_q2 <- variog(geo_areia_q2, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_areia_q2, main="Semivariograma areia da dicotomica para Q2")  
ajust_areia_q2 <- variofit(semi_areia_q2, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_areia_q2, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

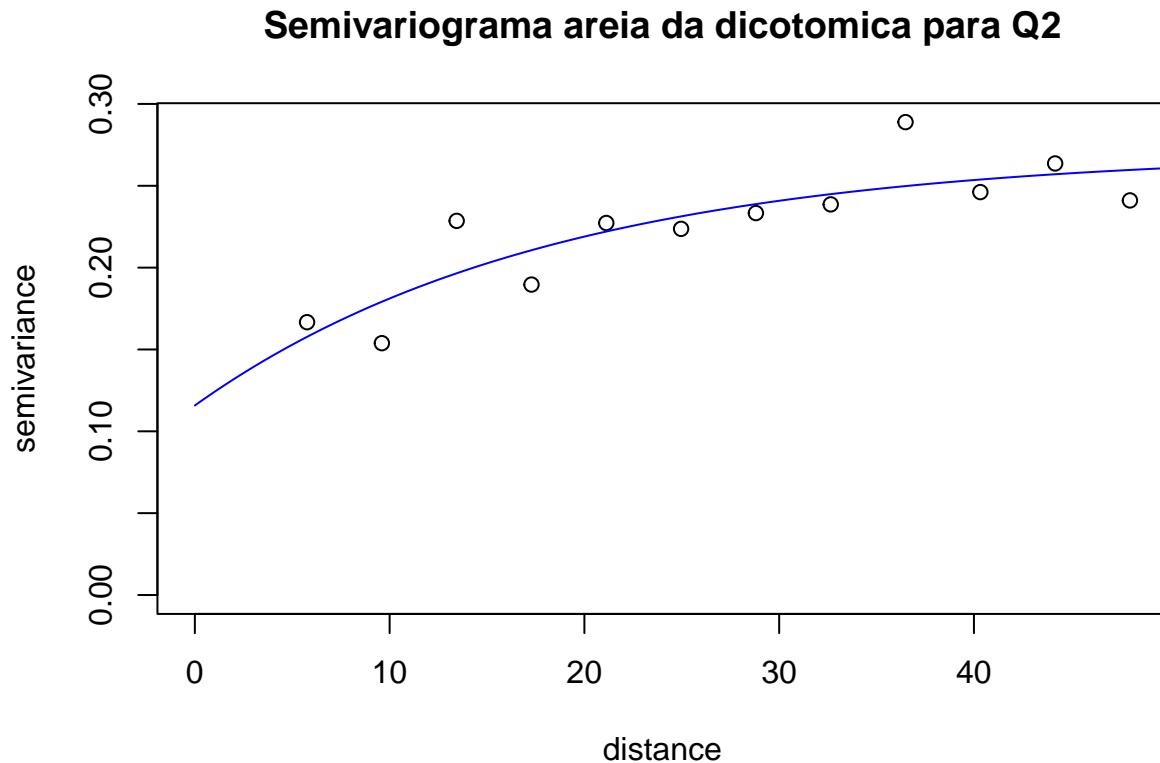
```
##           sigmasq   phi     tausq   kappa
```

```
## initial.value "0.14" "30.73" "0.14" "0.5"
```

```
## status       "est"    "est"    "est"    "fix"
```

```
## loss value: 0.00470925698196009
```

```
lines(ajust_areia_q2, col="blue")
```



```
semi_areia_md <- variog(geo_areia_md, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

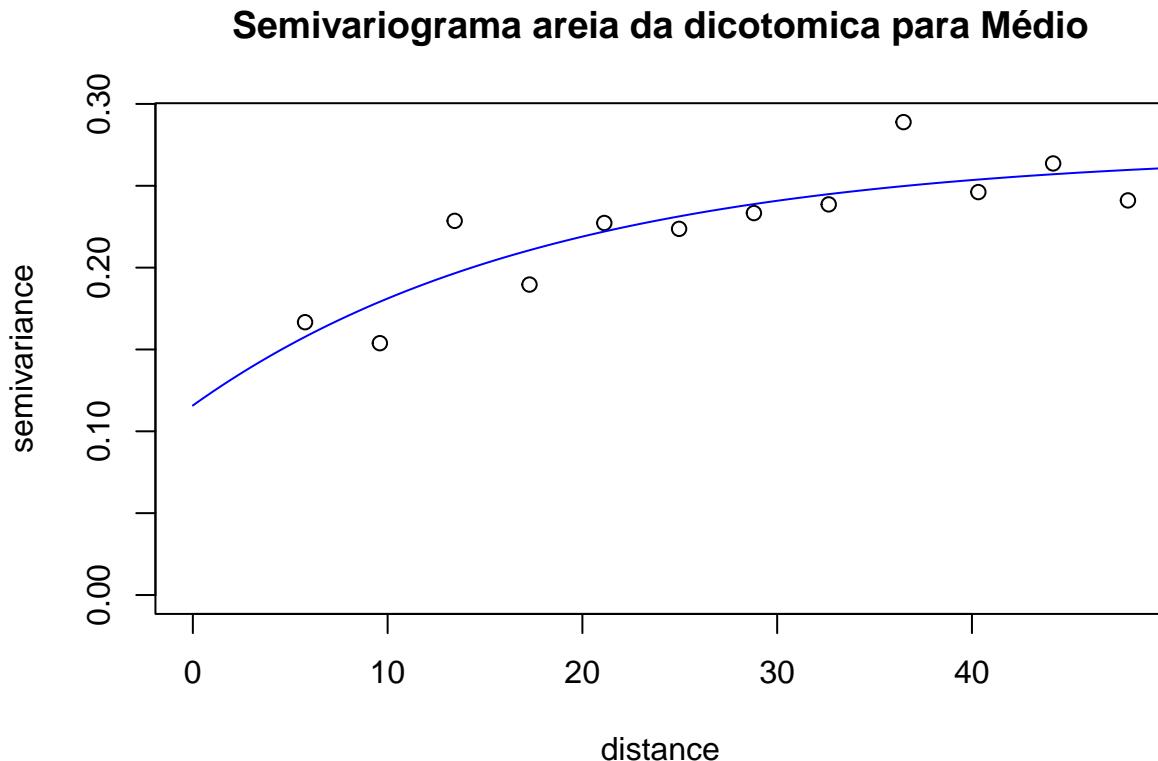
```
plot(semi_areia_md, main="Semivariograma areia da dicotomica para Médio")  
ajust_areia_md <- variofit(semi_areia_md, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern  
## variofit: weights used: equal  
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_areia_md, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:  
##           sigmasq   phi     tausq   kappa  
## initial.value "0.14" "30.73" "0.14" "0.5"  
## status        "est"   "est"   "est"   "fix"  
## loss value: 0.00470925698196009
```

```
lines(ajust_areia_md, col="blue")
```



```
semi_areia_q3 <- variog(geo_areia_q3, max.dist=50)
```

```
## variog: computing omnidirectional variogram
```

```
plot(semi_areia_q3, main="Semivariograma areia da dicotomica para Q3")  
ajust_areia_q3 <- variofit(semi_areia_q3, max.dist=50, wei="equal")
```

```
## variofit: covariance model used is matern
```

```
## variofit: weights used: equal
```

```
## variofit: minimisation function used: optim
```

```
## Warning in variofit(semi_areia_q3, max.dist = 50, wei = "equal"): initial values  
## not provided - running the default search
```

```
## variofit: searching for best initial value ... selected values:
```

```
##           sigmasq   phi    tausq   kappa
```

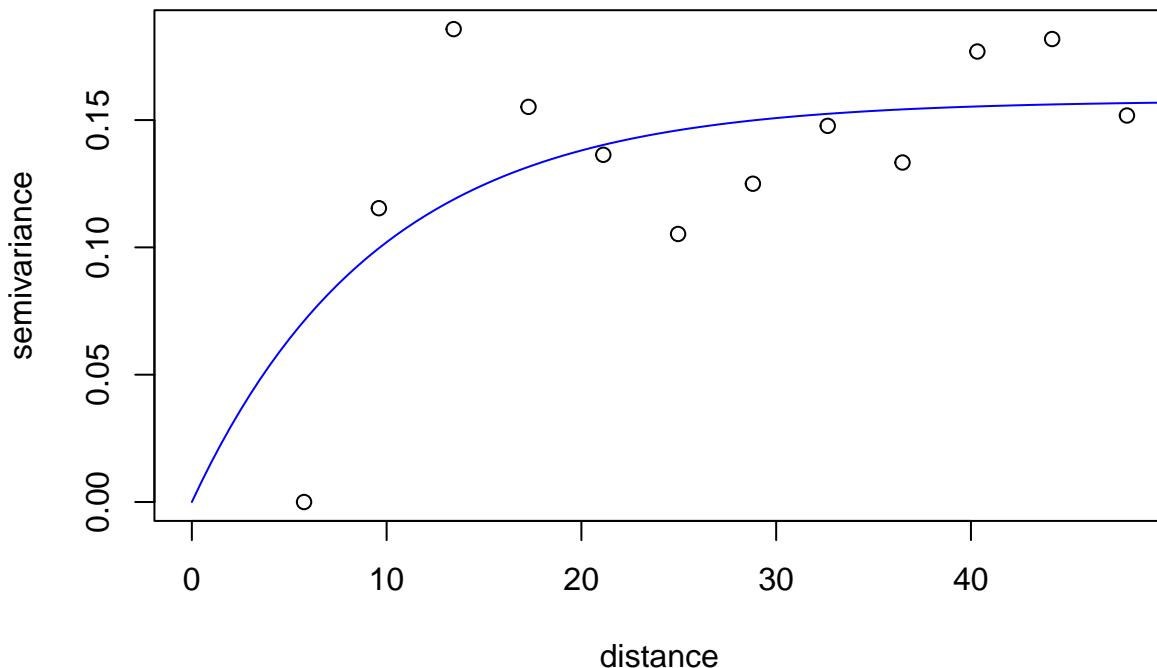
```
## initial.value "0.14"  "7.68"  "0"    "0.5"
```

```
## status        "est"   "est"   "est"  "fix"
```

```
## loss value: 0.0161701472048642
```

```
lines(ajust_areia_q3, col="blue")
```

Semivariograma areia da dicotomica para Q3



Krigagem da Indicadora

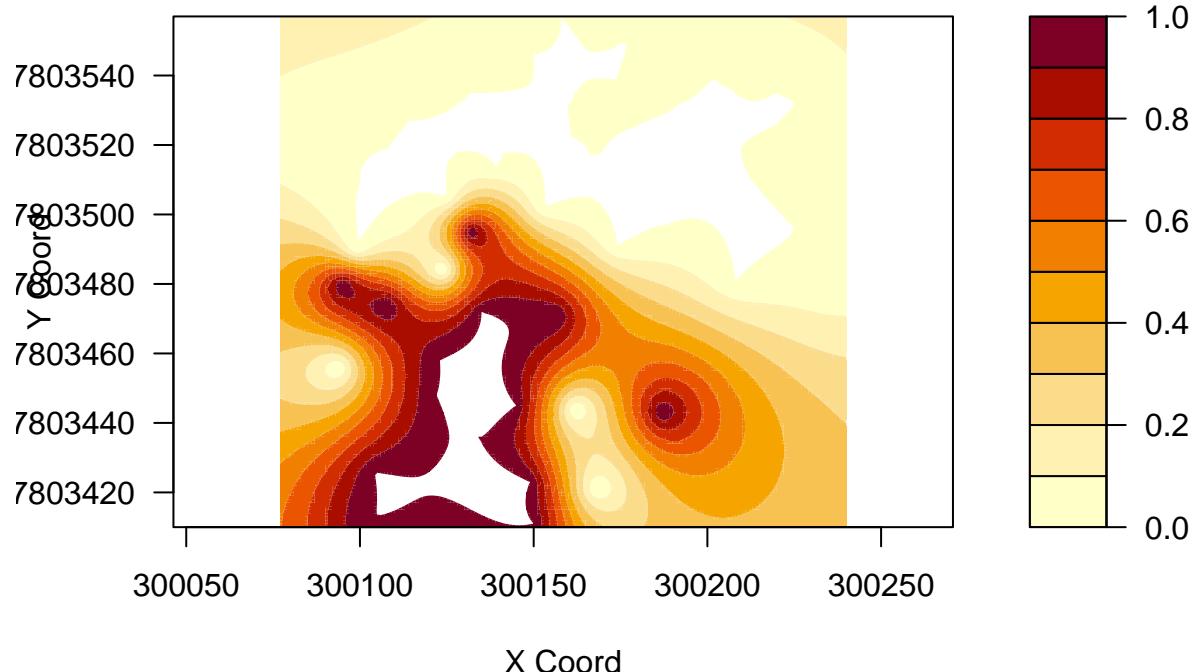
```
# definindo os locais para para as estimativas
loci <- expand.grid(seq(min(dados$Latitude),max(dados$Latitude),1), seq(min(dados$Longitude),max(dados$Longitude),1))

k = krige.conv(geo_altitude_q1, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_altitude))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

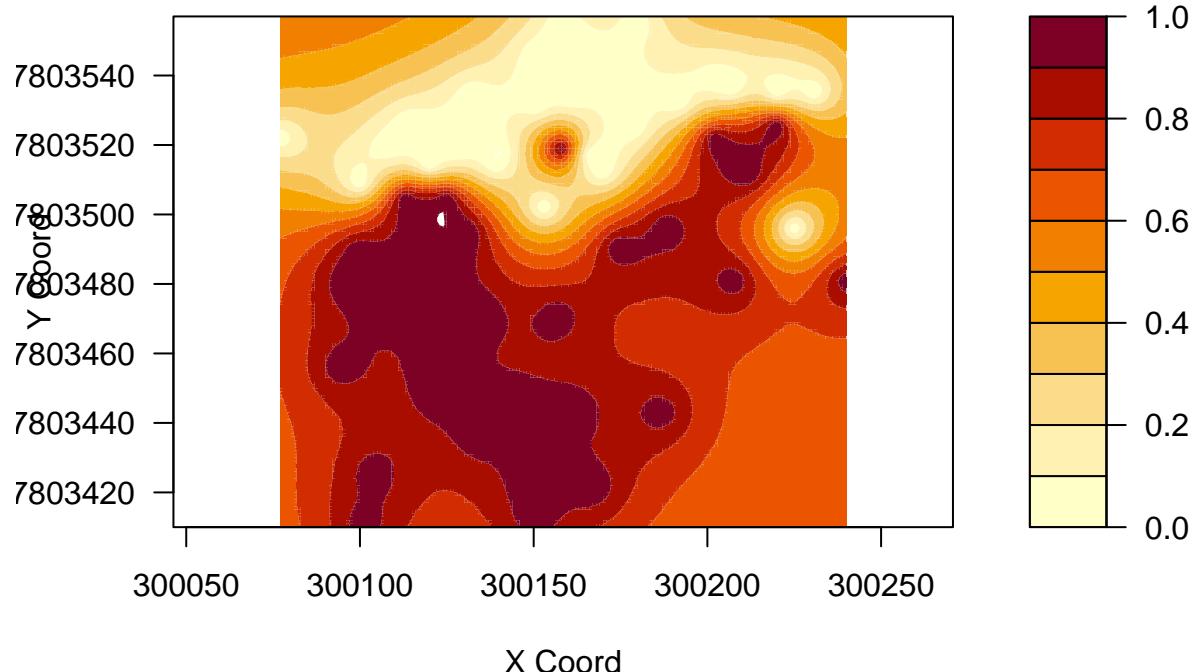
contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q1 para altitude")
```

Mapa de probabilidades menores ou igual a Q1



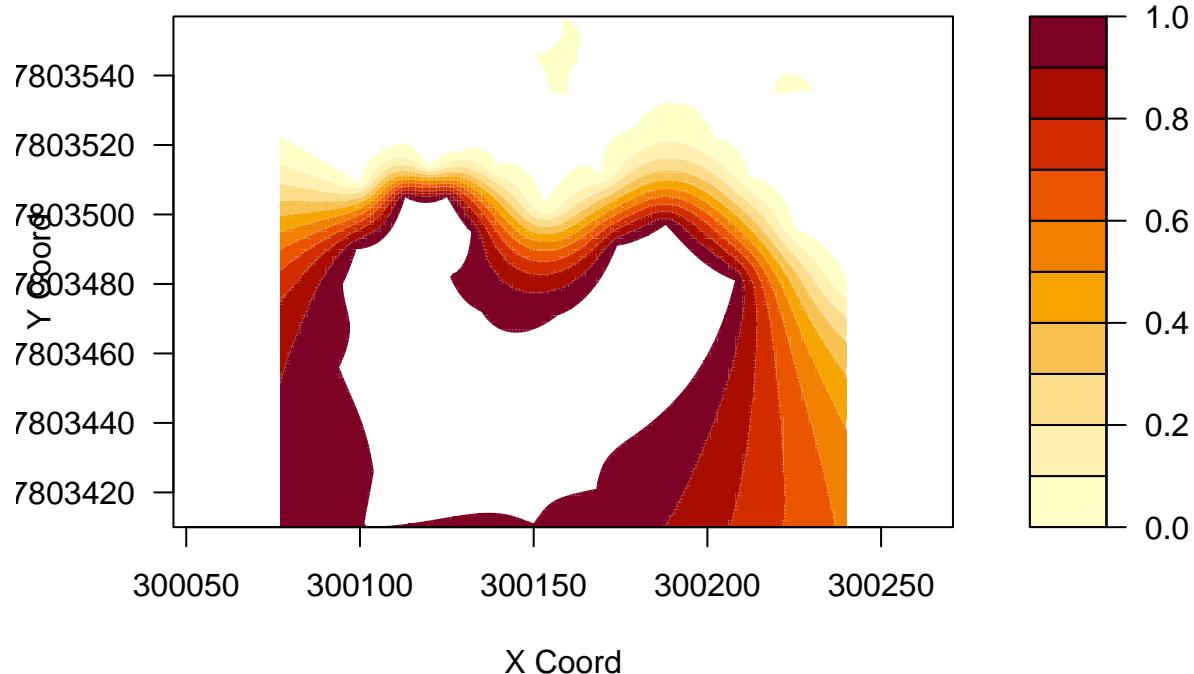
```
k = krige.conv(geo_altitude_q2, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_alti  
## krige.conv: model with constant mean  
## krige.conv: Kriging performed using global neighbourhood  
contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))  
title(main="Mapa de probabilidades menores ou igual a Q2 para altitude")
```

Mapa de probabilidades menores ou igual a Q2



```
k = krige.conv(geo_altitude_md, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_alti  
## krige.conv: model with constant mean  
## krige.conv: Kriging performed using global neighbourhood  
contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))  
title(main="Mapa de probabilidades menores ou igual a Médio para altitude")
```

Mapa de probabilidades menores ou igual a Média

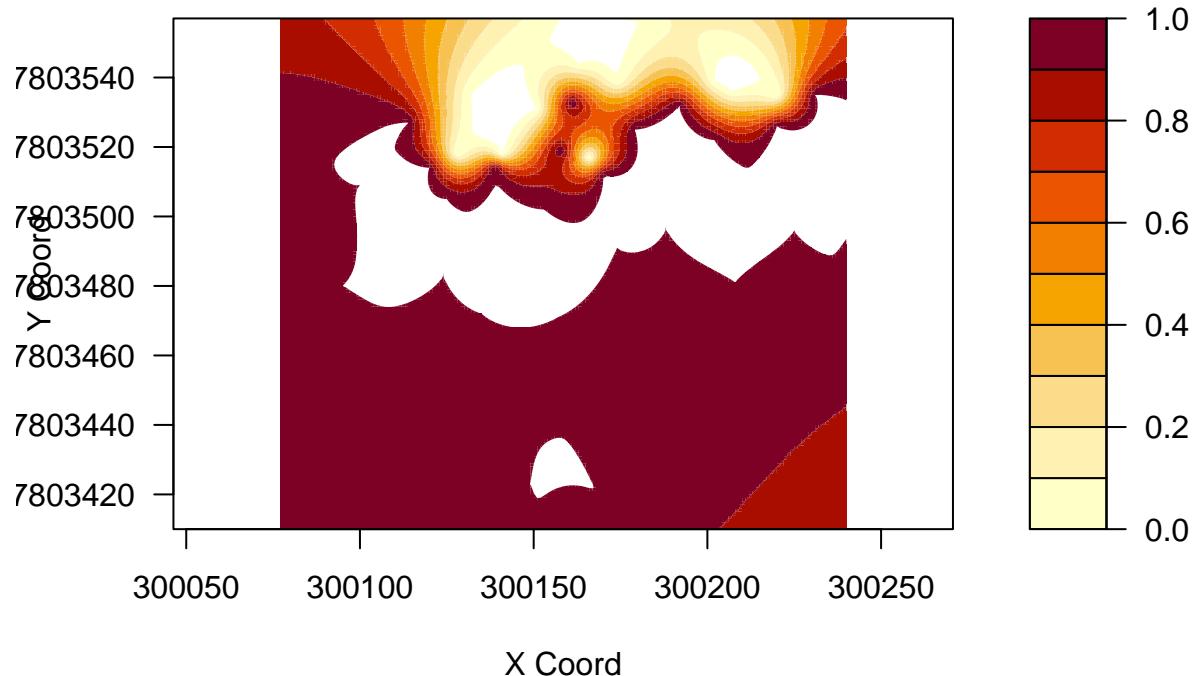


```
k = krige.conv(geo_altitude_q3, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_altitude))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q3 para altitude")
```

Mapa de probabilidades menores ou igual a Q3

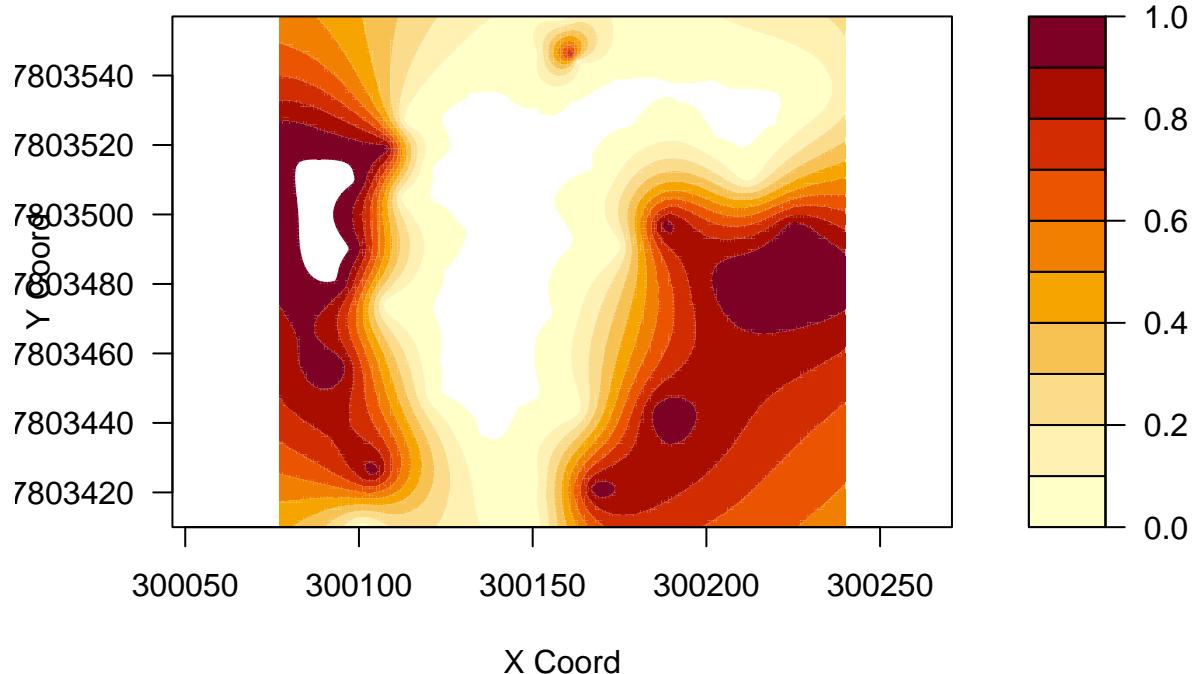


```
k = krige.conv(geo_umidade_q1, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_umidade))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q1 para umidade")
```

Mapa de probabilidades menores ou igual a Q1 p

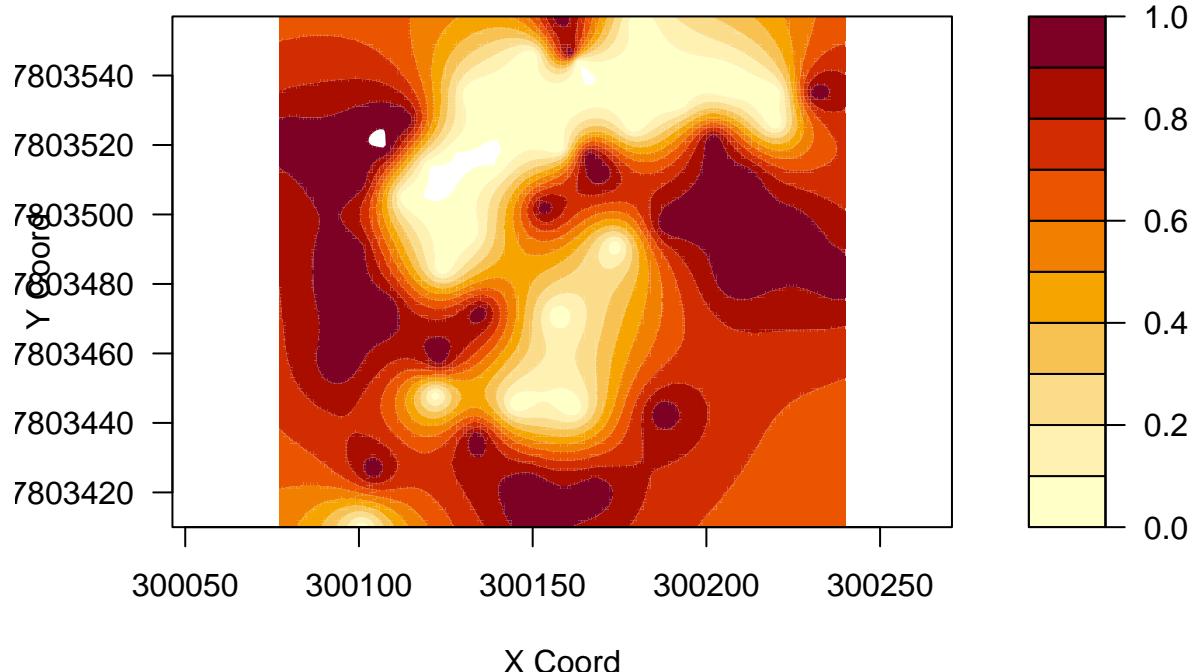


```
k = krige.conv(geo_umidade_q2, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_umidade))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q2 para umidade")
```

Mapa de probabilidades menores ou igual a Q2 p

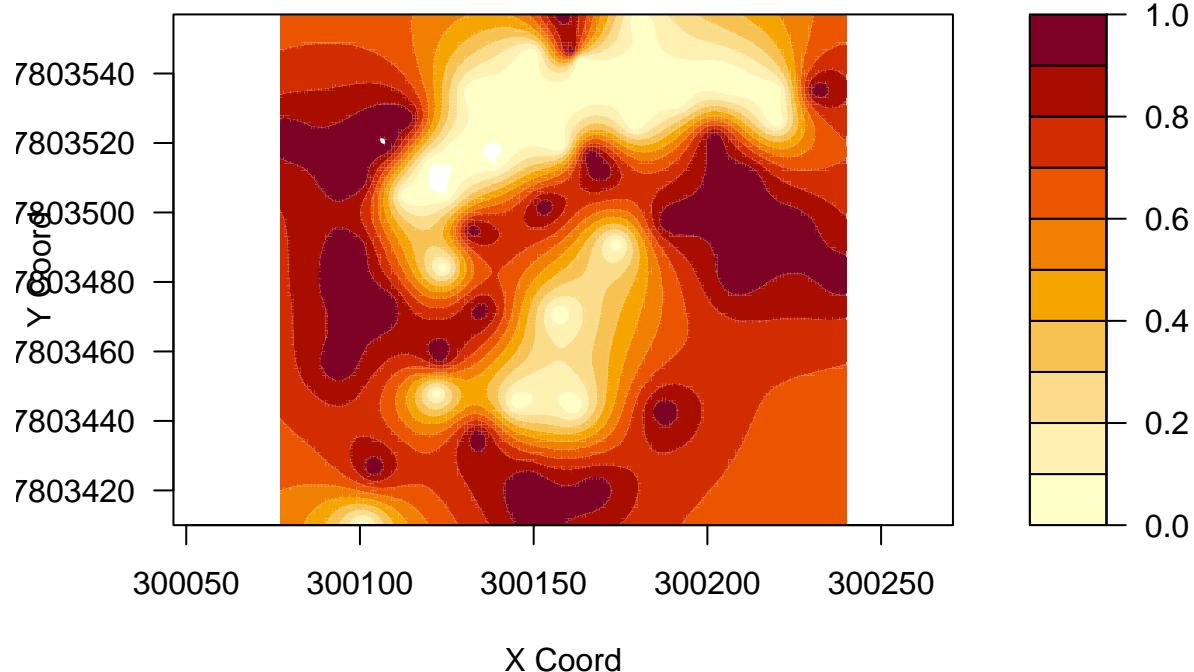


```
k = krige.conv(geo_umidade_md, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_umidade))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Médio para umidade")
```

Mapa de probabilidades menores ou igual a Média

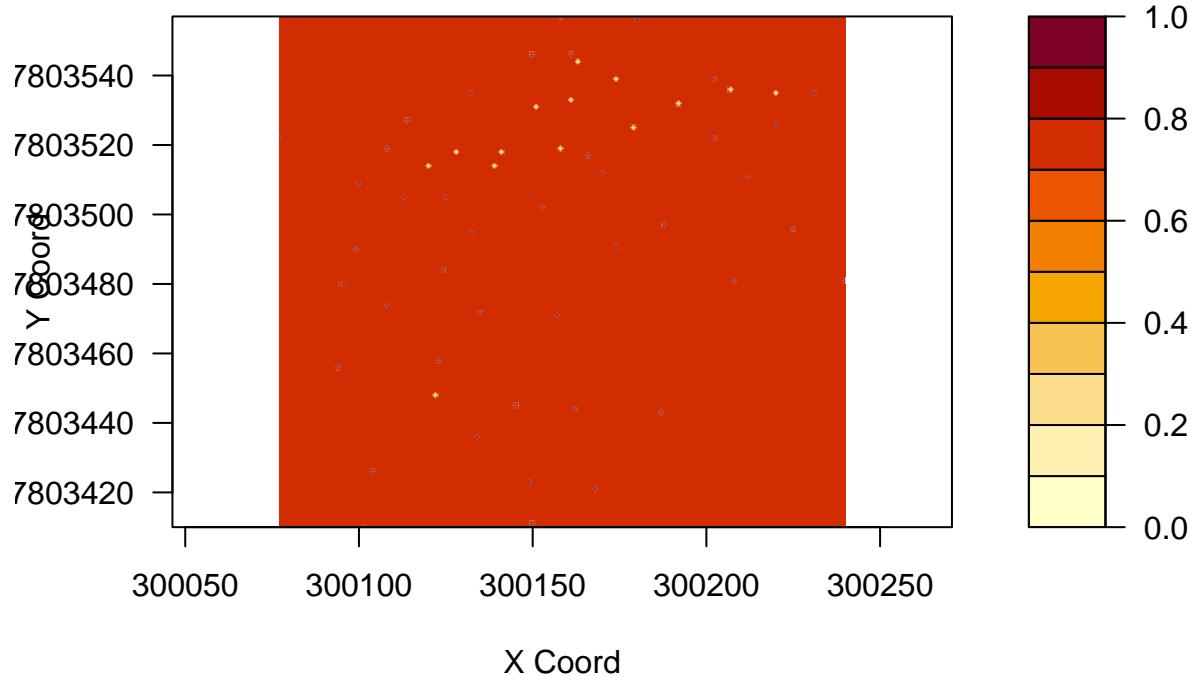


```
k = krige.conv(geo_umidade_q3, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_umidade))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q3 para umidade")
```

Mapa de probabilidades menores ou igual a Q3 p

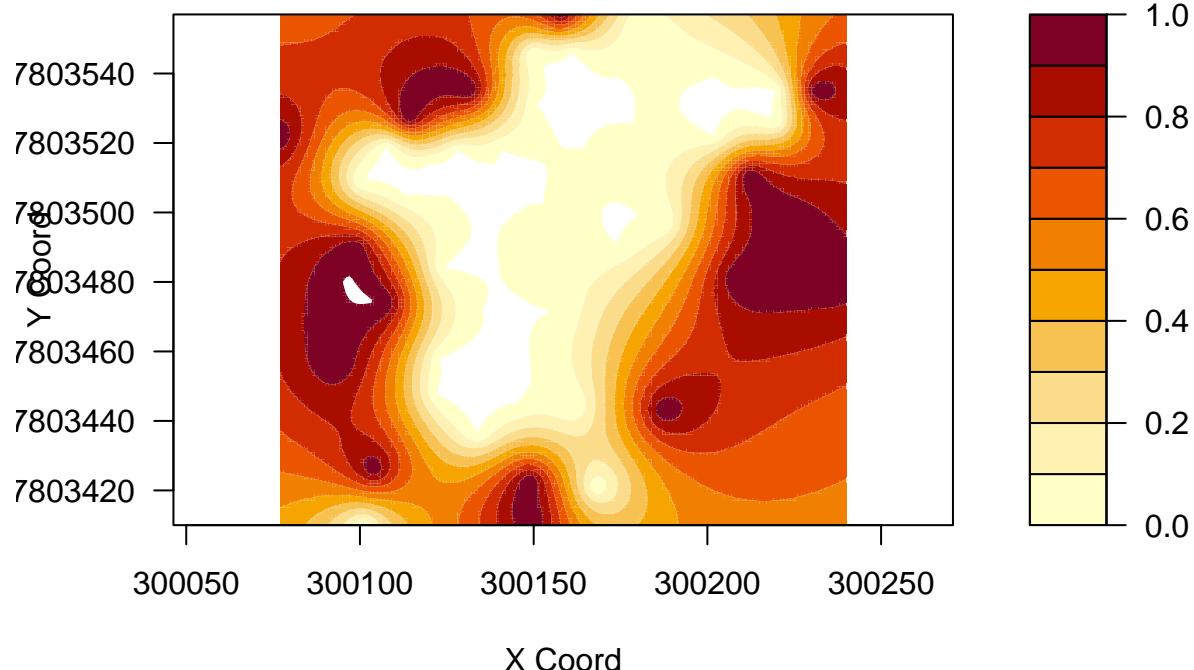


```
k = krige.conv(geo_argila_q1, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_argila))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q1 para argila")
```

Mapa de probabilidades menores ou igual a Q1

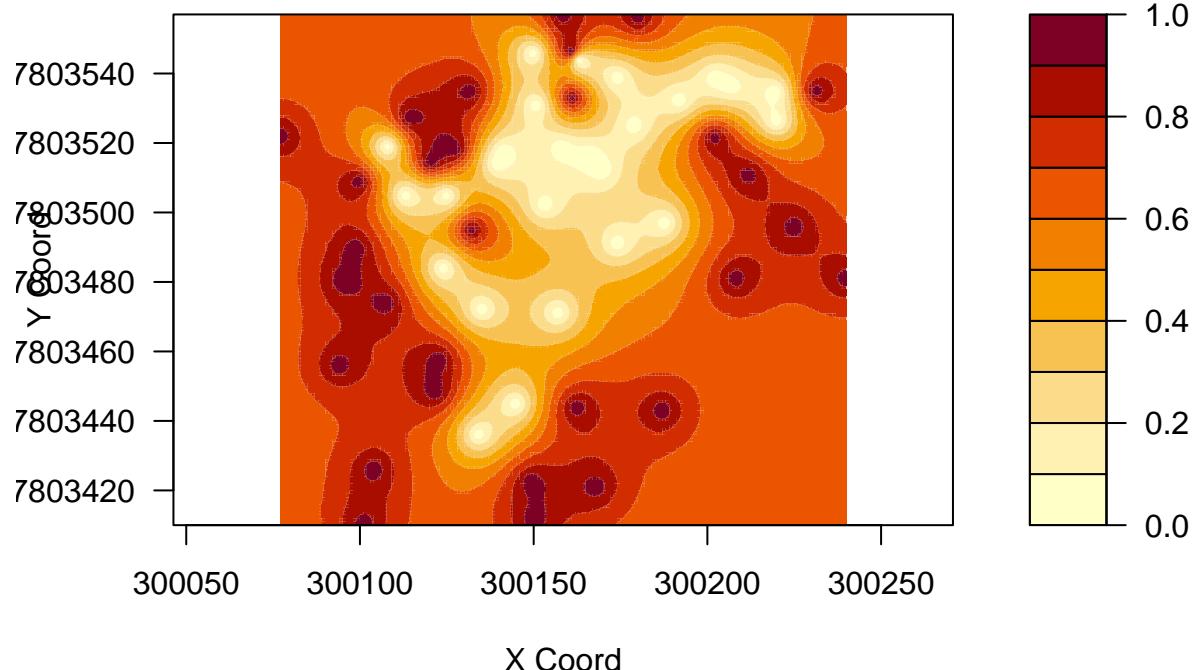


```
k = krige.conv(geo_argila_q2, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_argila))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q2 para argila")
```

Mapa de probabilidades menores ou igual a Q2

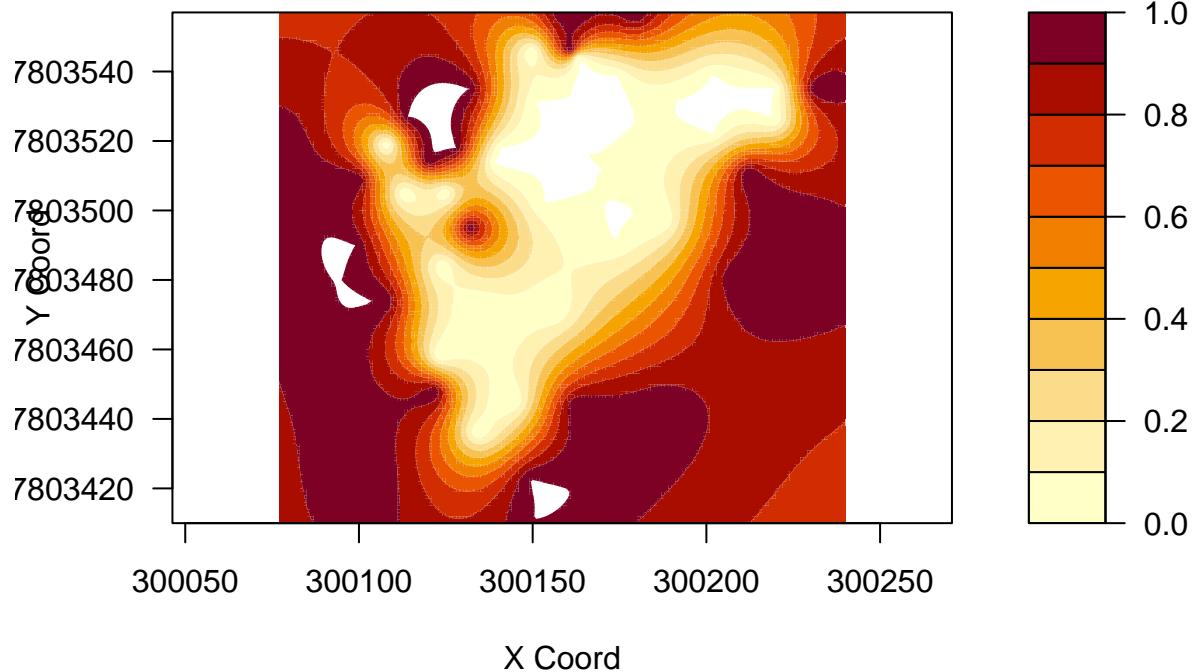


```
k = krige.conv(geo_argila_md, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_argila))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Médio para argila")
```

Mapa de probabilidades menores ou igual a Méd

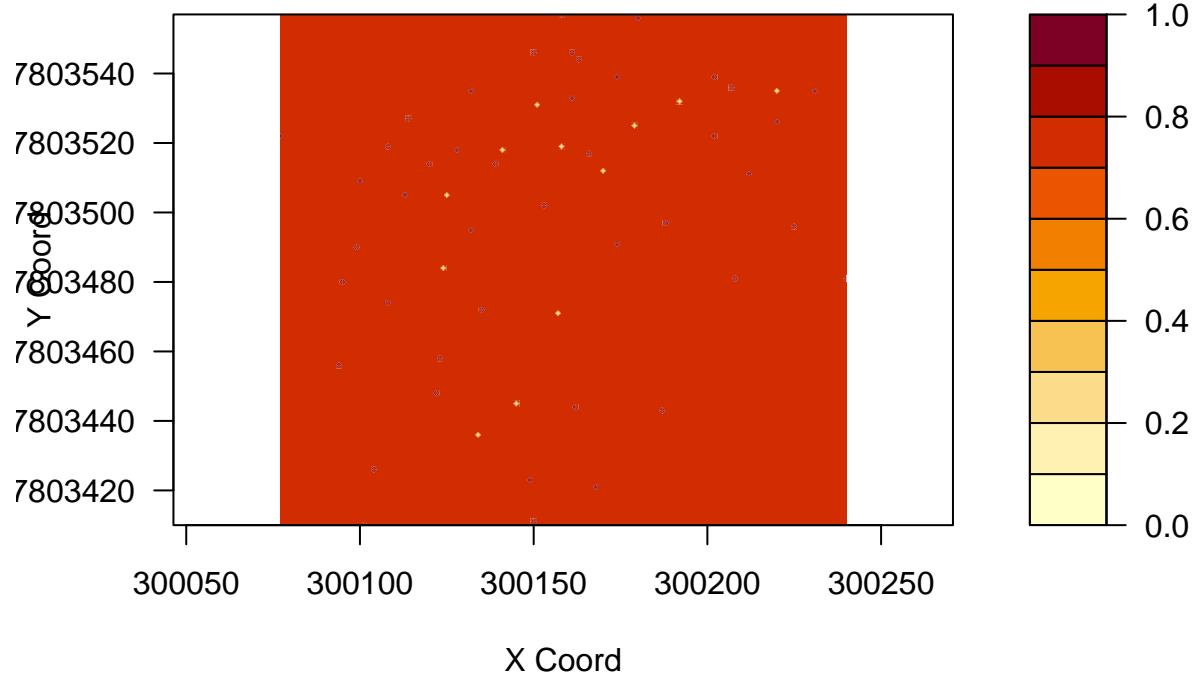


```
k = krige.conv(geo_argila_q3, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_argila))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q3 para argila")
```

Mapa de probabilidades menores ou igual a Q3

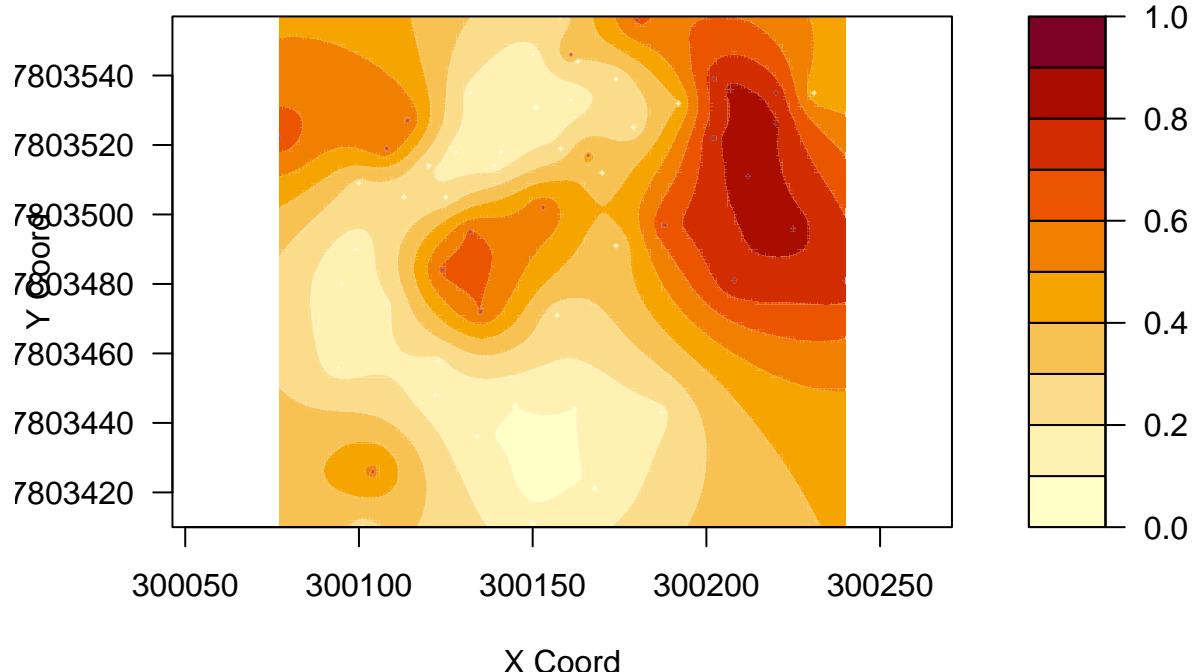


```
k = krige.conv(geo_silte_q1, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_silte_q1))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q1 para silte")
```

Mapa de probabilidades menores ou igual a Q²

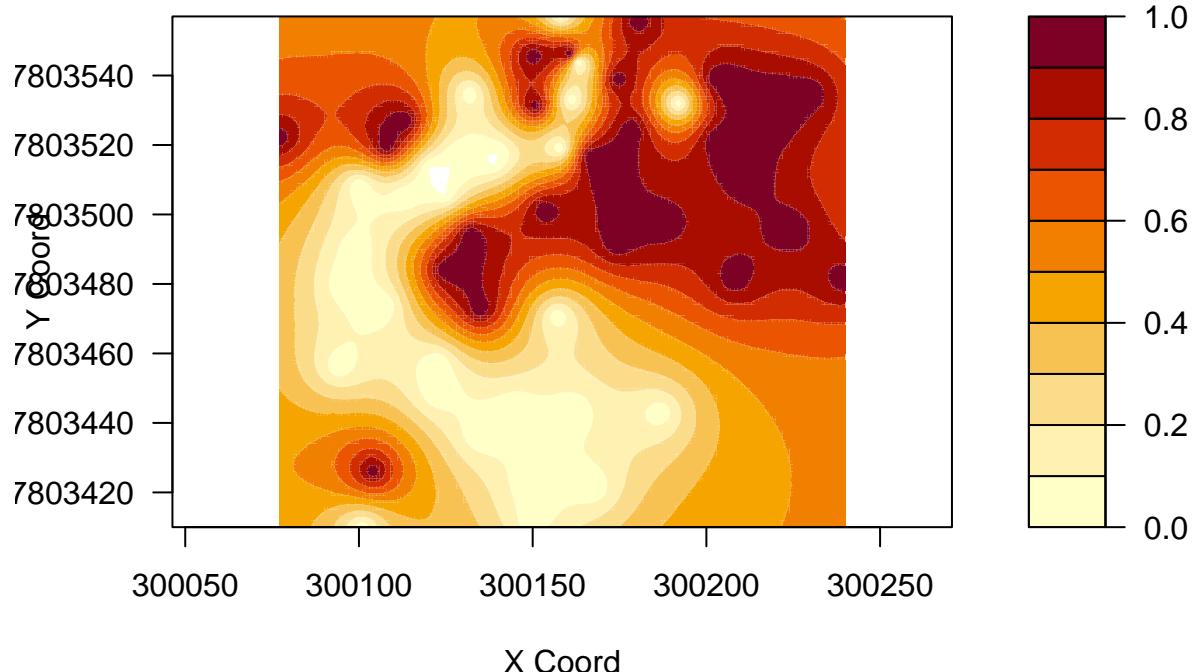


```
k = krige.conv(geo_silte_q2, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_silte_q2))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q2 para silte")
```

Mapa de probabilidades menores ou igual a Q_c

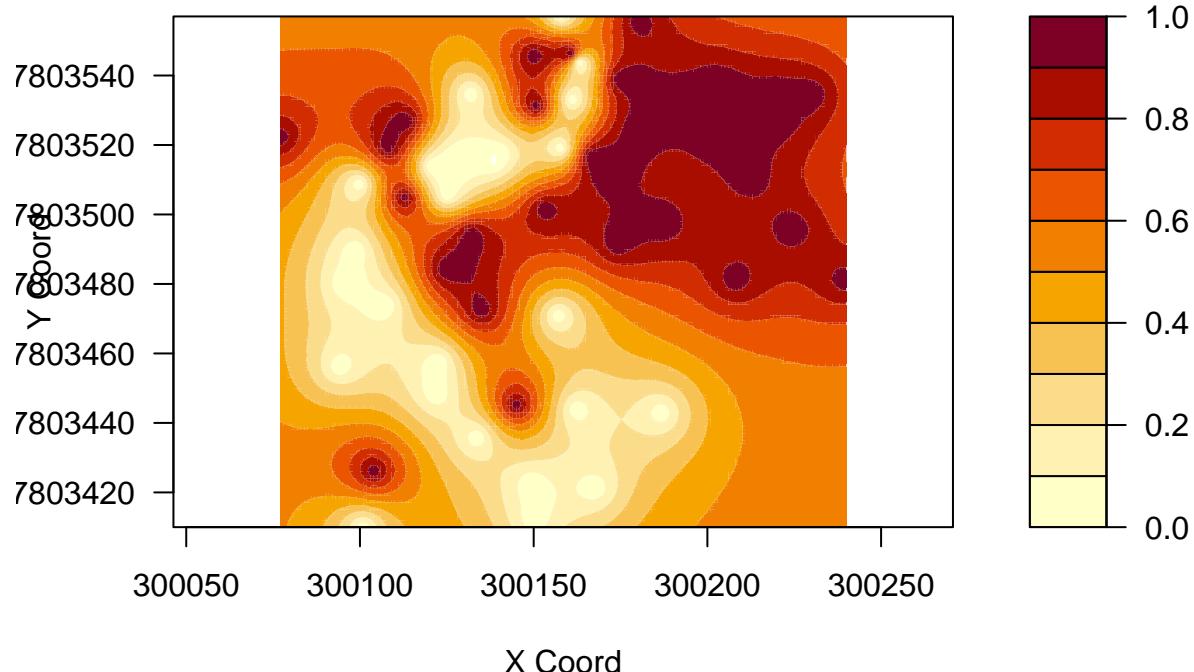


```
k = krige.conv(geo_silte_md, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_silte_md))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Médio para silte")
```

Mapa de probabilidades menores ou igual a Média

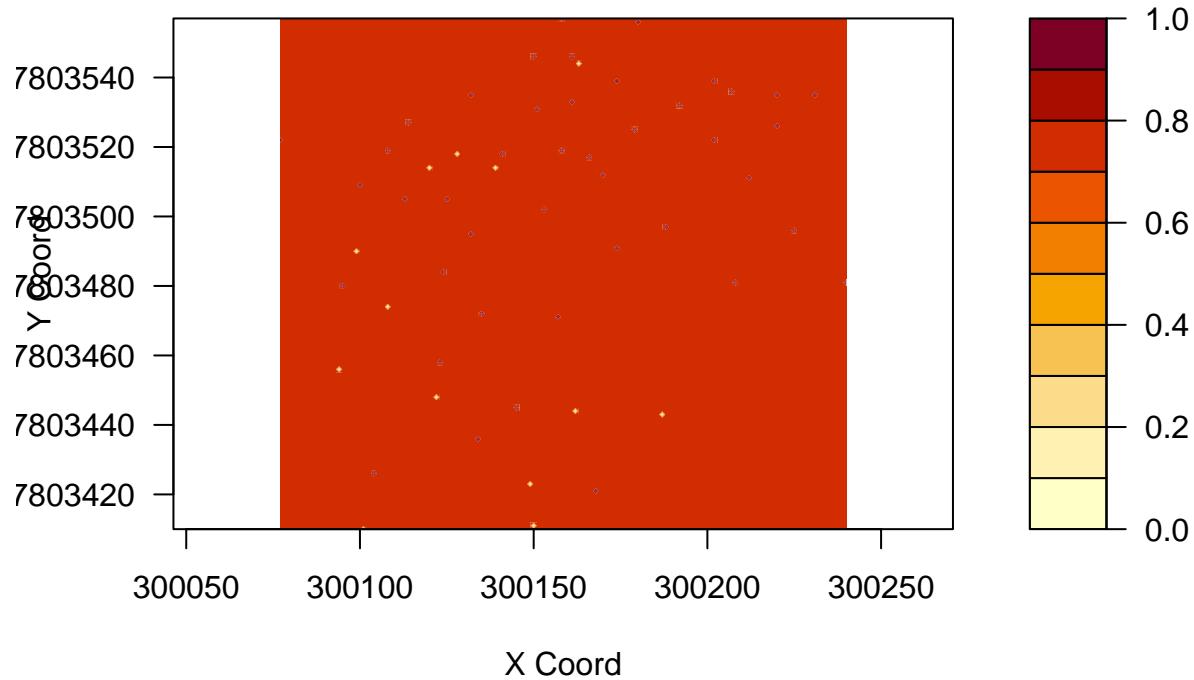


```
k = krige.conv(geo_silte_q3, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_silte_q3))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q3 para silte")
```

Mapa de probabilidades menores ou igual a Q₁



```
k = krige.conv(geo_areia_q1, loc=loci, krige=krige.control(type.krige = "ok", obj.model = ajust_areia_q1))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood

contour(k, filled=TRUE, levels=seq(0.00,1.00, by=0.1))
title(main="Mapa de probabilidades menores ou igual a Q1 para areia")
```

Mapa de probabilidades menores ou igual a Q1

