

Evaluación curso Introducción a la Geoestadística

Guzmán López

30 de noviembre de 2016

Contents

Consigna	1
Ambiente de trabajo en R	2
Información de la sesión	2
Librerías utilizadas	2
Cargar datos	3
Respuesta 1	4
Estructura y resumen de los datos	4
Descripción de la variación	5
Normalidad	11
Transformaciones	12
Q-Q Plots	15
Análisis de tendencias	19
Correlación	21
Autocorrelación espacial	31
Identificación de valores atípicos espaciales	33
Cambios en la media	47
Respuesta 2	49
Variograma empírico	49
Proceso espacial: isotropía y anisotropía	52
Respuesta 3	58
Ajustar modelos a variograma	59
Respuesta 4	63
Tamaño de la grilla y bloque	64
Kriging Ordinario Puntual	65
Kriging en Bloque	71
Respuesta 5	77
Validación cruzada	77
Respuesta 6	80
Simulaciones del proceso estocástico	80
Cálculo de probabilidad	81
Cálculo de superficie	83

Consigna

Se adjunta un archivo de datos espaciales de **rinde de maíz** y **covariables ambientales** para un lote de la región Pampeana para dos campañas agrícolas. El archivo está en formato de planilla de cálculo, con 2310 registros organizados en ocho campos/variables: Identificador, coordenadas

planares UTM20S, cota topográfica o elevación del terreno, profundidad de napa en floración para primera y segunda campaña (C1 y C2) y rendimiento de maíz en t/ha para la primera y segunda campaña. Los datos de rinde fueron obtenidos mediante monitores montados a cosechadoras y la profundidad de napa fue estimada a partir de la integración de datos en pozos de observación, modelos de simulación hidrodinámicos del acuífero y modelos digitales de elevación del terreno.

Ambiente de trabajo en R

En esta sección se establecieron y describen las características del ambiente de trabajo en R que fueron necesarias para poder comenzar a responder a las preguntas de la evaluación.

Información de la sesión

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Antergos Linux
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] RevoUtilsMath_10.0.0
##
## loaded via a namespace (and not attached):
## [1] backports_1.0.4  magrittr_1.5     RevoUtils_10.0.1 rprojroot_1.1
## [5] tools_3.3.1     htmltools_0.3.5  yaml_2.1.14     Rcpp_0.12.8
## [9] stringi_1.1.2   rmarkdown_1.2    knitr_1.15.1    stringr_1.1.0
## [13] digest_0.6.10   evaluate_0.10
```

Se utilizó el Ambiente de Desarrollo Integrado (*IDE*) de RStudio (versión 0.99.903) para R.

Librerías utilizadas

Se cargaron todas las librerías que fueron utilizadas para la realización de este trabajo.

```
# Geoestadística
library('gstat')
library('geoR')
library('geoRglm')

# Objetos espaciales
library('sp')

# Algunos estadísticos
library('moments')

# Reestructurar datos
library('reshape2')

# Transformaciones Box-Cox
library('forecast')
```

```

# Autocorrelaciones - Indices de Moran y Geary
library('ncf')

# Dependencia espacial
library('spdep')

# Akaike
library('MASS')

# Importancia relativa predictores
library('relaimpo')

# Gráficos

# Versión dev de ggplot2, requiere:
# devtools::install_github("hadley/scales")
# devtools::install_github("tidyverse/ggplot2")
library('ggplot2')
library('gridExtra')
library('corrgram') # correlogramas

# Escalas de colores
library('RColorBrewer')

```

Cargar datos

Se cargaron en R los datos espaciales de **rinde de maíz** y **covariables ambientales** para un lote de la región Pampeana para dos campañas agrícolas. Para ello, el archivo original en formato *Hoja de cálculo de Microsoft Excel* (DataExamenGeo16.xlsx) fue convertido a un *Documento de texto plano* (DataExamenGeo16.csv). Para la conversión se mantuvieron las cabeceras originales, se asignó como delimitador de campos la coma (,) y el punto (.) como separador decimal y se asignó una extensión de archivo de tipo .csv (valores separados comas).

```

# Cargar datos
datosRindeMaiz <- read.csv(file = "data/DataExamenGeo16.csv", header = TRUE,
                           sep = ",", dec = ".")  
  

# Mostrar las primeras seis filas
head(datosRindeMaiz)  
  

##   ID      Lat     Long   Elev NapaDicC1 NapaDicC2  maizC1 maizC2
## 1  1 3913.263 16153.52 245.99      4.98      4.38  6.8833 4.1816
## 2  2 3933.332 16153.53 245.90      4.85      4.25  9.1300 6.4079
## 3  3 3953.291 16153.54 245.59      4.50      3.90  7.6300 6.2962
## 4  4 3973.250 16153.56 245.48      4.34      3.74  7.5800 6.9209
## 5  5 3993.319 16153.57 245.34      4.16      3.56  9.3425 7.3573
## 6  6 4013.278 16153.58 245.23      4.01      3.41 10.4800 7.1703

```

Los datos **rendimiento de maíz** (*maizC1* y *maizC2*) y **covariables ambientales** (*Elev*, *NapaDicC1* y *NapaDicC2*) fueron de tipo variables continuas. Mientras que el espacio fue de dos dimensiones dado por $x = \{Long, Lat\}$.

Crear objeto espacial (`SpatialPointsDataFrame`)

El objeto `datosRindeMaiz` de tipo `data.frame` fue transformado a un objeto espacial de tipo `SpatialPointsDataFrame` asignando las columnas `Long` y `Lat` como las coordenadas del objeto.

```
coordinates(datosRindeMaiz) <- c("Long", "Lat")
```

Asignar Sistema de Coordenadas de Referencia (CRS)

Dado que se utilizó un Sistema de Coordenadas de Referencia relativo, se asignó al objeto `datosRindeMaiz` un Sistema de Coordenadas de Referencia vacío.

```
proj4string(datosRindeMaiz) <- CRS(as.character(NA))
```

Respuesta 1

- 1) Realice un análisis descriptivo exhaustivo para evaluar los supuestos distribucionales del rinde para ambas campañas, la existencia de cambios en la media dentro del lote y la presencia de valores atípicos. Justifique todas las decisiones que tome respecto de los datos originales.

Estructura y resumen de los datos

El objeto `datosRindeMaiz` posee:

1. un objeto de clase `data.frame` llamado `data` con los atributos de los datos de rinde de maíz sin las coordenadas (2310 observaciones y 6 variables: `ID`, `Elev`, `NapaDicC1`, `NapaDicC2`, `maizC1`, `maizC2`)
2. un objeto de clase `integer` llamado `coords.nrs` con dos valores que significan el número de columna del `data.frame` original del que se tomaron los datos de las coordenadas
3. un objeto de clase `matrix` llamado `coords` de 2310 filas por 2 columnas con las coordenadas (`Long` y `Lat`)
4. un objeto de clase `matrix` llamado `bbox` de 2 filas por 2 columnas con los límites espaciales (`Long min`, `Long max`, `Lat min` y `Lat max`)
5. un objeto de clase `CRS` llamado `proj4string` con el Sistema de Coordenadas de Referencia

```
str(datosRindeMaiz)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
## ..@ data      : 'data.frame': 2310 obs. of 6 variables:
## ...$ ID       : int [1:2310] 1 2 3 4 5 6 7 8 9 10 ...
## ...$ Elev     : num [1:2310] 246 246 246 245 245 ...
## ...$ NapaDicC1: num [1:2310] 4.98 4.85 4.5 4.34 4.16 4.01 3.93 3.8 3.69 3.59 ...
## ...$ NapaDicC2: num [1:2310] 4.38 4.25 3.9 3.74 3.56 3.41 3.33 3.2 3.09 2.99 ...
## ...$ maizC1   : num [1:2310] 6.88 9.13 7.63 7.58 9.34 ...
## ...$ maizC2   : num [1:2310] 4.18 6.41 6.3 6.92 7.36 ...
## ..@ coords.nrs : int [1:2] 3 2
## ..@ coords    : num [1:2310, 1:2] 16154 16154 16154 16154 16154 ...
## ...- attr(*, "dimnames")=List of 2
## ...  ..$ : NULL
## ...  ..$ : chr [1:2] "Long" "Lat"
## ..@ bbox      : num [1:2, 1:2] 16154 3913 17233 4733
## ...- attr(*, "dimnames")=List of 2
## ...  ..$ : chr [1:2] "Long" "Lat"
## ...  ..$ : chr [1:2] "min" "max"
```

```
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
## ... . . . @ projargs: chr NA
```

Descripción de la variación

Distribución de frecuencias

Para ilustrar como la distribución de los datos de cada variable se ubica con respecto a su mediana o para identificar valores extremos se realizaron Histogramas y Diagramas de cajas para cada una de las variables.

```
# Ver histogramas de las variables
ggDatosRindeMaiz <- ggplot(data = datosRindeMaiz$data)

# Maíz C1
ggHistMaizC1 <- ggDatosRindeMaiz +
  geom_histogram(aes(x = datosRindeMaiz$maizC1),
                 binwidth = 0.5, fill = "orange", col = "white", alpha = 0.8) +
  labs(title = "", x = "Rendimiento maíz (t/ha)", y = "Frecuencia") +
  scale_x_continuous(limits = c(0, 18), breaks = seq(0, 18, 2)) +
  scale_y_continuous(limits = c(0, 325), breaks = seq(0, 325, 50)) +
  theme(plot.margin = unit(c(-1, 1, 1, 1), "mm"))

ggBoxPlotMaizC1 <- ggDatosRindeMaiz +
  geom_boxplot(aes(y = datosRindeMaiz$maizC1,
                   x = rep("RM", length(datosRindeMaiz$maizC1))),
               fill = "orange", col = "black", alpha = 0.8) +
  labs(title = "C1", x = "", y = "") +
  scale_y_continuous(limits = c(0, 18), breaks = seq(0, 18, 2), position = "top") +
  theme(plot.margin = unit(c(1, 1, -1, 1), "mm")) +
  coord_flip()

# Napa C1
ggHistNapaDicC1 <- ggDatosRindeMaiz +
  geom_histogram(aes(x = datosRindeMaiz$NapaDicC1),
                 binwidth = 0.25, fill = "darkblue", col = "white", alpha = 0.7) +
  labs(title = "", x = "Profundidad napa (m)", y = "Frecuencia") +
  scale_x_continuous(limits = c(1.5, 6.75), breaks = seq(1.5, 6.75, 1)) +
  scale_y_continuous(limits = c(0, 350), breaks = seq(0, 350, 50)) +
  theme(plot.margin = unit(c(-1, 1, 1, 1), "mm"))

ggBoxPlotNapaDicC1 <- ggDatosRindeMaiz +
  geom_boxplot(aes(y = datosRindeMaiz$NapaDicC1,
                   x = rep("PN", length(datosRindeMaiz$NapaDicC1))),
               fill = "darkblue", col = "black", alpha = 0.7) +
  labs(title = "C1", x = "", y = "") +
  scale_y_continuous(limits = c(1.5, 6.75),
                     breaks = seq(1.5, 6.75, 1), position = "top") +
  theme(plot.margin = unit(c(1, 1, -1, 1), "mm")) +
  coord_flip()

# Maíz C2
ggHistMaizC2 <- ggDatosRindeMaiz +
  geom_histogram(aes(x = datosRindeMaiz$maizC2),
                 binwidth = 0.5, fill = "orange", col = "white", alpha = 0.8) +
```

```

  labs(title = "", x = "Rendimiento maíz (t/ha)", y = "Frecuencia") +
  scale_x_continuous(limits = c(0, 18), breaks = seq(0, 18, 2)) +
  scale_y_continuous(limits = c(0, 325), breaks = seq(0, 325, 50)) +
  theme(plot.margin = unit(c(-1, 1, 1, 1), "mm"))

ggBoxPlotMaizC2 <- ggDatosRindeMaiz +
  geom_boxplot(aes(y = datosRindeMaiz$maizC2,
                    x = rep("RM", length(datosRindeMaiz$maizC2))),
                fill = "orange", col = "black", alpha = 0.8) +
  labs(title = "C2", x = "", y = "") +
  scale_y_continuous(limits = c(0, 18),
                     breaks = seq(0, 18, 2), position = "top") +
  theme(plot.margin = unit(c(1, 1, -1, 1), "mm")) +
  coord_flip()

# Napa C2
ggHistNapaDicC2 <- ggDatosRindeMaiz +
  geom_histogram(aes(x = datosRindeMaiz$NapaDicC2),
                 binwidth = 0.25, fill = "darkblue", col = "white",
                 alpha = 0.7) +
  labs(title = "", x = "Profundidad napa (m)", y = "Frecuencia") +
  scale_x_continuous(limits = c(1.5, 6.75), breaks = seq(1.5, 6.75, 1)) +
  scale_y_continuous(limits = c(0, 350), breaks = seq(0, 350, 50)) +
  theme(plot.margin = unit(c(-1, 1, 1, 1), "mm"))

ggBoxPlotNapaDicC2 <- ggDatosRindeMaiz +
  geom_boxplot(aes(y = datosRindeMaiz$NapaDicC2,
                    x = rep("PN", length(datosRindeMaiz$NapaDicC2))),
                fill = "darkblue", col = "black", alpha = 0.7) +
  labs(title = "C2", x = "", y = "") +
  scale_y_continuous(limits = c(1.5, 6.75),
                     breaks = seq(1.5, 6.75, 1), position = "top") +
  theme(plot.margin = unit(c(1, 1, -1, 1), "mm")) +
  coord_flip()

# Elevación
ggHistElev <- ggDatosRindeMaiz +
  geom_histogram(aes(x = datosRindeMaiz$Elev),
                 binwidth = 0.2, fill = "darkred", col = "white", alpha = 0.7) +
  labs(title = "", x = "Elevación (m)", y = "Frecuencia") +
  scale_x_continuous(limits = c(241.5, 247.5), breaks = seq(240, 248, 1)) +
  scale_y_continuous(limits = c(0, 275), breaks = seq(0, 275, 50)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.margin = unit(c(-1, 1, 1, 1), "mm"))

ggBoxPlotElev <- ggDatosRindeMaiz +
  geom_boxplot(aes(y = datosRindeMaiz$Elev,
                    x = rep("EL", length(datosRindeMaiz$Elev))),
                fill = "darkred", col = "black", alpha = 0.7) +
  labs(title = "", x = "", y = "") +
  scale_y_continuous(limits = c(241.5, 247.5),
                     breaks = seq(240, 248, 1), position = "top") +
  theme(plot.margin = unit(c(1, 1, -1, 1), "mm"))

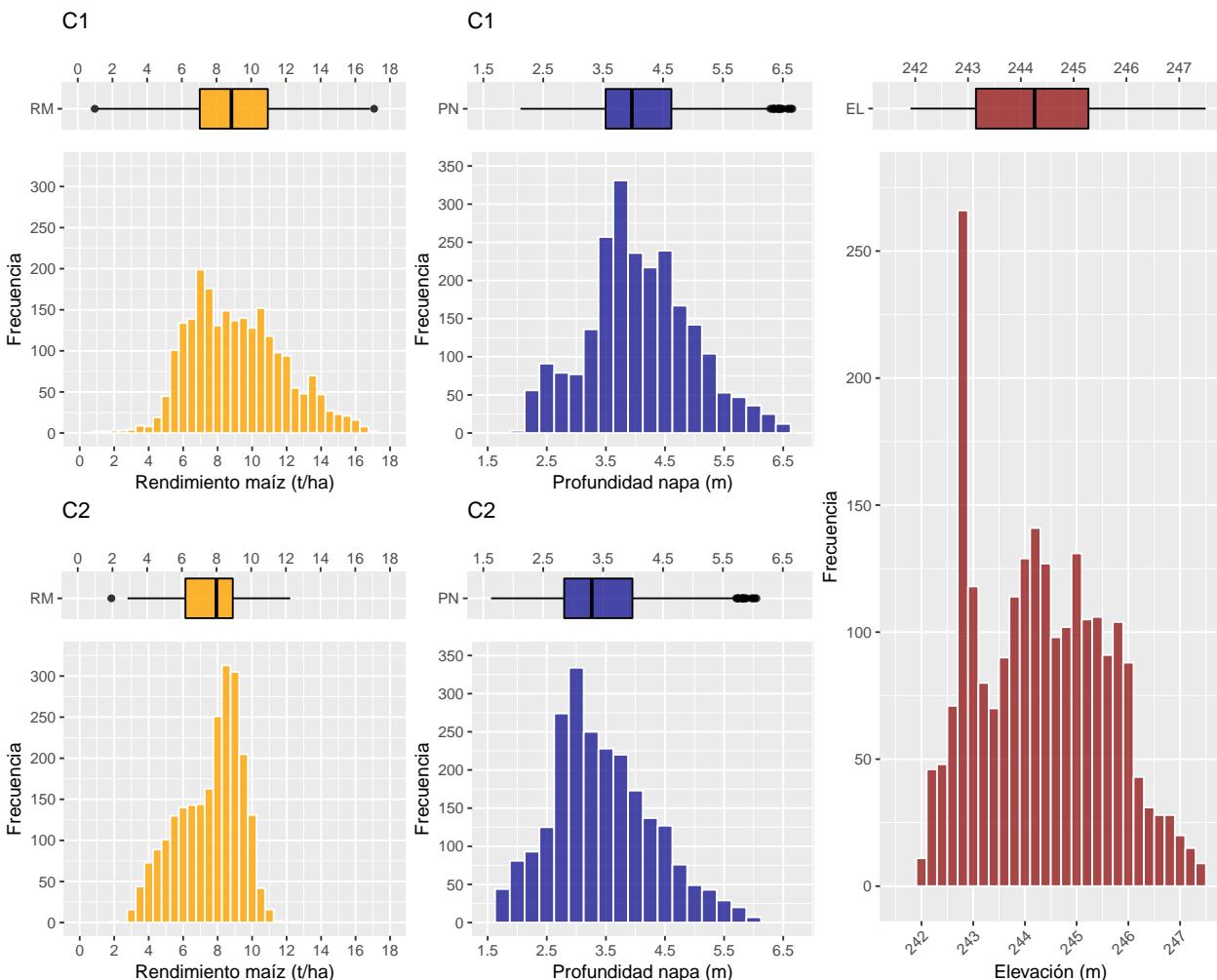
```

```

coord_flip()

# Ver plot
grid.arrange(ggBoxPlotMaizC1,
             ggHistMaizC1,
             ggBoxPlotMaizC2,
             ggHistMaizC2,
             ggBoxPlotNapaDicC1,
             ggHistNapaDicC1,
             ggBoxPlotNapaDicC2,
             ggHistNapaDicC2,
             ggBoxPlotElev,
             ggHistElev,
             ncol = 3, nrow = 4,
             layout_matrix = cbind(c(1,2,3,4), c(5,6,7,8), c(9,10,10,10)),
             widths = c(1, 1, 1), heights = c(1, 3, 1, 3))

```



Se observó para los datos de *rendimiento de maíz* que el valor de la mediana fue mayor para la *campaña 1* que para la *campaña 2* y que la dispersión de valores entorno a la mediana fue mayor para la *campaña 1* que para la *campaña 2*. Por otro lado, la frecuencia de valores de *rendimiento de maíz* por debajo de la mediana fue mayor para la *campaña 2* que para la *campaña 1* y la frecuencia de valores de *rendimiento de maíz* por encima de la mediana fue mayor para la *campaña 1* que para la *campaña 2*. Se observaron valores extremos

de *rendimiento de maíz* mayores y menores a la mediana para la *campaña 1* y solo menores a la mediana para la *campaña 2*.

Para los datos de la *profundidad de napa* se observó que el valor de la mediana fue mayor para la *campaña 1* que para la *campaña 2* y que la dispersión de valores entorno a la mediana y la frecuencia de valores mayores y menores a la mediana fue semejante para ambas campañas. En ambas campañas se observaron valores extremos mayores a la mediana.

Para los datos de *elevación* se observó dispersión de valores entorno a la mediana y la frecuencia entre valores mayores y menores a la mediana fue semejante. No se observaron valores extremos para la *elevación*.

Tendencia central y dispersión

Para analizar cuantitativamente el rango, la tendencia central y la dispersión de los datos de *rendimiento de maíz* y covariables ambientales se creó una función personalizada llamada **ResumenEstadisticas**. Además se incluyó que dicha función retorne el tipo de clase de la variable en R y la cantidad de celdas vacías (NA).

```
ResumenEstadisticas <- function(df) {  
  
  # Número de columnas del data.frame  
  ncol = ncol(df)  
  
  # Estadísticas  
  maximo = numeric()  
  minimo = numeric()  
  
  media = numeric()  
  mediana = numeric()  
  moda = vector()  
  
  varianza = numeric()  
  desvioEstandar = numeric()  
  cv = numeric()  
  asimetria = numeric()  
  kurtosis = numeric()  
  
  NAs = numeric()  
  clases = character()  
  
  # Función para calcular la moda  
  Moda <- function(x) {  
    ux = unique(x)  
    ux[which.max(tabulate(match(x, ux)))]  
  }  
  
  # Iteración que recorre todas las columnas del data.frame  
  for(i in 1:ncol) {  
  
    # Condición de que la columna para las estadísticas tenga que ser numérica  
    if(is.numeric(df[,i])) {  
  
      mediana[i] = median(df[,i], na.rm = TRUE)  
      media[i] = mean(df[,i], na.rm = TRUE)  
      moda[i] = Moda(x = df[,i])  
      varianza[i] = var(df[,i], na.rm = TRUE)  
    }  
  }  
}
```

```

desvioEstandar[i] = sd(df[,i], na.rm = TRUE)
cv[i] = 100 * (sd(df[,i], na.rm = TRUE) / mean(df[,i], na.rm = TRUE))
asimetria[i] = moments::skewness(df[,i], na.rm = TRUE)
kurtosis[i] = moments::kurtosis(df[,i], na.rm = TRUE)
minimo[i] = min(df[,i], na.rm = TRUE)
maximo[i] = max(df[,i], na.rm = TRUE)
NAs[i] = length(which(is.na(df[,i]) == TRUE))
clases[i] = class(df[,i])

}

# En caso de que no sea de tipo numérica asignar NAs a las
# estadísticas pero calcular el número de NAs y tipo de datos
else {

  mediana[i] = NA
  media[i] = NA
  moda[i] = Moda(x = df[,i])
  varianza[i] = NA
  desvioEstandar[i] = NA
  cv[i] = 100 * NA
  asimetria[i] = NA
  kurtosis[i] = NA
  minimo[i] = NA
  maximo[i] = NA
  NAs[i] = length(which(is.na(df[,i]) == TRUE))
  clases[i] = class(df[,i])

}

}

# Generar data.frame con todas las estadísticas
summary = data.frame("mínimo" = minimo,
                      "máximo" = maximo,
                      "media" = media,
                      "mediana" = mediana,
                      "moda" = moda,
                      "varianza" = varianza,
                      "ds" = desvioEstandar,
                      "cv" = cv,
                      "asimetría" = asimetria,
                      "kurtosis" = kurtosis,
                      "NAs" = NAs,
                      "clase" = clases)

# Asignar el nombre de las variables (columnas)
# como nombre de las filas para el objeto de salida
rownames(summary) <- colnames(df)

# Retorno de la función
return(summary)
}

```

Se llamó a la función `ResumenEstadisticas` pasando como parámetro el `data.frame` llamado `data` contenido en el objeto espacial `datosRindeMaiz` y se asignó el retorno de la función a un nuevo objeto llamado `resumenDatosRindeMaiz`. Además, se imprimió en consola el resultado de la ejecución del comando.

```
# Crear objeto con las principales estadísticas y mostrarlo en consola
(resumenDatosRindeMaiz <- ResumenEstadisticas(df = datosRindeMaiz@data))
```

```
##          mínimo     máximo      media   mediana    moda    varianza
## ID        1.0000 2310.0000 1155.50000 1155.50000  1.000 4.448675e+05
## Elev     241.9100  247.5000  244.310848 244.26000 242.770 1.560964e+00
## NapaDicC1  2.1200   6.6600   4.070797  3.98000  3.690 7.859209e-01
## NapaDicC2  1.6300   6.0600   3.442294  3.31000  3.030 7.373578e-01
## maizC1    0.9800  17.0850   9.134139  8.85935  7.000 7.433428e+00
## maizC2    1.9381  12.2521   7.552832  7.99375  9.402 3.309716e+00
##           ds       cv  asimetría kurtosis  NAs    clase
## ID        666.9838829 57.7225342 0.0000000 1.800000  0 integer
## Elev     1.2493853  0.5113916  0.2129448 2.135257  0 numeric
## NapaDicC1  0.8865218 21.7776002  0.1903320 2.875718  0 numeric
## NapaDicC2  0.8586954 24.9454385  0.4559732 2.906922  0 numeric
## maizC1    2.7264314 29.8488072  0.3713567 2.693631  2 numeric
## maizC2    1.8192624 24.0871547 -0.5174745 2.397373  0 numeric
```

Marcar celdas vacías

```
NAs <- which(is.na(datosRindeMaiz$maizC1))
```

A través del objeto `resumenDatosRindeMaiz` fue posible obtener el/los estadístico/s necesarios. Por ejemplo:

1. Obtener el valor de la mediana del Rendimiento del maíz para la Campaña 1

Ejemplo 1

```
(resumenDatosRindeMaiz["maizC1", "mediana"])
```

```
## [1] 8.85935
```

2. Obtener los valores de mínimo y máximo de la Elevación

Ejemplo 2

```
(resumenDatosRindeMaiz["Elev", c("mínimo", "máximo")])
```

```
##          mínimo     máximo
## Elev  241.91  247.5
```

Se observaron dos celdas vacías para los datos de *rendimiento de maíz* de la *campaña 2*. Todas las variables fueron de tipo numéricas a excepción de la variable identificadora *ID* que fue un entero.

Tendencia central

Los valores de la media (\bar{z}) y la mediana (Md) del *rendimiento de maíz* (Rm) para la *campaña 1* fueron mayores que para la *campaña 2* ($\bar{z}_{Rm1} = 9.13 \text{ t.ha}^{-1}$, $Md_{Rm1} = 8.86 \text{ t.ha}^{-1}$ y $\bar{z}_{Rm2} = 7.55 \text{ t.ha}^{-1}$, $Md_{Rm2} = 7.99 \text{ t.ha}^{-1}$) mientras que el valor de la moda (Mo) fue de $Mo_{Rm1} = 7.00 \text{ t.ha}^{-1}$ y $Mo_{Rm1} = 9.40 \text{ t.ha}^{-1}$ respectivamente.

Los valores de la media y la mediana de la *profundidad de la napa* (Pn) para la *campaña 1* ($\bar{z}_{Pn1} = 4.07 \text{ m}$, $Md_{Pn1} = 3.98 \text{ m}$) fueron mayores que para la *campaña 2* ($\bar{z}_{Pn2} = 3.44 \text{ m}$, $Md_{Pn2} = 3.31 \text{ m}$) mientras que el valor de la moda fue de 3.69 m y 3.03 m respectivamente.

Los valores de la media, la mediana y la moda para la *elevación* fueron de $\bar{z}_{Ele} = 244.31 \text{ m}$, $Md_{Ele} = 244.26 \text{ m}$ y $Mo_{Ele} = 242.77 \text{ m}$ respectivamente.

Dispersión

El rango ($R = x_{min}, x_{max}$) de la variable del *rendimiento de maíz* para la *campaña 1* fue mayor y con valores más altos y más bajos que para la *campaña 2* ($R_{Rm1} = 0.9800 \text{ t.ha}^{-1}$, $17.0850 \text{ t.ha}^{-1}$, $R_{Rm2} = 1.9381 \text{ t.ha}^{-1}$, $12.2521 \text{ t.ha}^{-1}$), para la variable *profundidad de napa* el rango fue similar para las dos campañas ($R_{Pn1} = 2.12 \text{ m}$, 6.66 m , $R_{Pn2} = 1.63 \text{ m}$, 6.06 m) y el rango de la *elevación* fue de $R_{Ele} = 241.91 \text{ m}$, 247.50 m .

Normalidad

Para analizar la normalidad de las variables se realizó la prueba de normalidad de *Shapiro-Wilk*. La hipótesis nula de la prueba de *Shapiro-Wilk* es que los datos son normales, por lo cual un *p-valor* significativo indicó que se rechazó la hipótesis nula, es decir que la variable analizada fuese normal.

```
# Elevation
shapiro.test(datosRindeMaiz$Elev)

##
## Shapiro-Wilk normality test
##
## data: datosRindeMaiz$Elev
## W = 0.97116, p-value < 2.2e-16

# Profundidad de la napa (C1)
shapiro.test(datosRindeMaiz$NapaDicC1)

##
## Shapiro-Wilk normality test
##
## data: datosRindeMaiz$NapaDicC1
## W = 0.99124, p-value = 1.297e-10

# Profundidad de la napa (C2)
shapiro.test(datosRindeMaiz$NapaDicC2)

##
## Shapiro-Wilk normality test
##
## data: datosRindeMaiz$NapaDicC2
## W = 0.98135, p-value < 2.2e-16

# Rendimiento del maíz (C1)
shapiro.test(datosRindeMaiz$maizC1)

##
## Shapiro-Wilk normality test
##
## data: datosRindeMaiz$maizC1
## W = 0.98251, p-value = 3.07e-16

# Rendimiento del maíz (C2)
shapiro.test(datosRindeMaiz$maizC2)

##
## Shapiro-Wilk normality test
##
## data: datosRindeMaiz$maizC2
## W = 0.95925, p-value < 2.2e-16
```

Para el *rendimiento de maíz* y las covariables ambientales se rechazó la hipótesis nula de la prueba de normalidad de *Shapiro-Wilk*.

Transformaciones

Se transformó los datos originales a nuevas escalas para acercarlos a una distribución normal. Se probaron las transformaciones *logarítmica*, *raíz cuadrada* y *Box-Cox*.

```
# Transformar elevación

# Logaritmo
ElevLog <- log(datosRindeMaiz@data$Elev)
shapiro.test(ElevLog) # Probar normalidad de la transformación

## 
## Shapiro-Wilk normality test
##
## data: ElevLog
## W = 0.97134, p-value < 2.2e-16

# Raíz cuadrada
ElevSqrt <- sqrt(datosRindeMaiz@data$Elev)
shapiro.test(ElevSqrt) # Probar normalidad de la transformación

## 
## Shapiro-Wilk normality test
##
## data: ElevSqrt
## W = 0.97125, p-value < 2.2e-16

# Box-Cox
lambdaElev <- BoxCox.lambda(x = datosRindeMaiz@data$Elev) # Hallar Lambda
ElevBC <- BoxCox(x = datosRindeMaiz@data$Elev, lambda = lambdaElev) # Transf.vector
shapiro.test(ElevBC) # Probar normalidad de la transformación

## 
## Shapiro-Wilk normality test
##
## data: ElevBC
## W = 0.9715, p-value < 2.2e-16

# Transformar profundidad napa (C1)

# Logaritmo
NapaDicC1Log <- log(datosRindeMaiz@data$NapaDicC1)
shapiro.test(NapaDicC1Log)

## 
## Shapiro-Wilk normality test
##
## data: NapaDicC1Log
## W = 0.98123, p-value < 2.2e-16

# Raíz cuadrada
NapaDicC1Sqrt <- sqrt(datosRindeMaiz@data$NapaDicC1)
shapiro.test(NapaDicC1Sqrt)
```

```

##  

## Shapiro-Wilk normality test  

##  

## data: NapaDicC1Sqrt  

## W = 0.99189, p-value = 4.568e-10  

# Box-Cox  

lambdaNapaDicC1 <- BoxCox.lambda(x = datosRindeMaiz@data$NapaDicC1)  

NapaDicC1BC <- BoxCox(x = datosRindeMaiz@data$NapaDicC1, lambda = lambdaNapaDicC1)  

shapiro.test(NapaDicC1BC)

##  

## Shapiro-Wilk normality test  

##  

## data: NapaDicC1BC  

## W = 0.9875, p-value = 2.42e-13  

# Transformar profundidad napa (C2)

# Logaritmo  

NapaDicC2Log <- log(datosRindeMaiz@data$NapaDicC2)  

shapiro.test(NapaDicC2Log)

##  

## Shapiro-Wilk normality test  

##  

## data: NapaDicC2Log  

## W = 0.99209, p-value = 6.856e-10  

# Raíz cuadrada  

NapaDicC2Sqrt <- sqrt(datosRindeMaiz@data$NapaDicC2)  

shapiro.test(NapaDicC2Sqrt)

##  

## Shapiro-Wilk normality test  

##  

## data: NapaDicC2Sqrt  

## W = 0.99313, p-value = 6.067e-09  

# Box-Cox  

lambdaNapaDicC2 <- BoxCox.lambda(x = datosRindeMaiz@data$NapaDicC2)  

NapaDicC2BC <- BoxCox(x = datosRindeMaiz@data$NapaDicC2, lambda = lambdaNapaDicC2)  

shapiro.test(NapaDicC2BC)

##  

## Shapiro-Wilk normality test  

##  

## data: NapaDicC2BC  

## W = 0.99138, p-value = 1.709e-10  

# Transformar rendimiento maíz (C1)

# Logaritmo  

maizC1Log <- log(datosRindeMaiz@data$maizC1)  

shapiro.test(maizC1Log)

##  

## Shapiro-Wilk normality test

```

```

##  

## data: maizC1Log  

## W = 0.96474, p-value < 2.2e-16  

# Raíz cuadrada  

maizC1Sqrt <- sqrt(datosRindeMaiz$data$maizC1)  

shapiro.test(maizC1Sqrt)

##  

## Shapiro-Wilk normality test  

##  

## data: maizC1Sqrt  

## W = 0.99035, p-value = 2.61e-11  

# Box-Cox  

lambdaMaizC1Log <- BoxCox.lambda(x = datosRindeMaiz$data$maizC1)  

maizC1BC <- BoxCox(x = datosRindeMaiz$data$maizC1, lambda = lambdaMaizC1Log)  

shapiro.test(maizC1BC)

##  

## Shapiro-Wilk normality test  

##  

## data: maizC1BC  

## W = 0.9473, p-value < 2.2e-16  

# Transformar rendimiento maíz (C2)

# Logaritmo  

maizC2Log <- log(datosRindeMaiz$data$maizC2)  

shapiro.test(maizC2Log)

##  

## Shapiro-Wilk normality test  

##  

## data: maizC2Log  

## W = 0.90957, p-value < 2.2e-16  

# Raíz cuadrada  

maizC2Sqrt <- sqrt(datosRindeMaiz$data$maizC2)  

shapiro.test(maizC2Sqrt)

##  

## Shapiro-Wilk normality test  

##  

## data: maizC2Sqrt  

## W = 0.93893, p-value < 2.2e-16  

# Box-Cox  

lambdaMaizC2Log <- BoxCox.lambda(x = datosRindeMaiz$data$maizC2)  

maizC2BC <- BoxCox(x = datosRindeMaiz$data$maizC2, lambda = lambdaMaizC2Log)  

shapiro.test(maizC2BC)

##  

## Shapiro-Wilk normality test  

##  

## data: maizC2BC  

## W = 0.96354, p-value < 2.2e-16

```

Ninguna de las transformaciones de los datos permitió que se pudiera fallar en rechazar la hipótesis nula de normalidad del test de *Shapiro-Wilk*.

Q-Q Plots

```
# Q-Q plot rendimiento maíz (C1)
ggQQmaizC1 <- ggDatosRindeMaiz +
  geom_qq(aes(sample = maizC1)) +
  geom_abline(intercept = mean(datosRindeMaiz@data$maizC1, na.rm = TRUE),
              slope = sd(datosRindeMaiz@data$maizC1, na.rm = TRUE), col = "red") +
  labs(title = "Rend.maíz C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC1Log <- ggDatosRindeMaiz +
  geom_qq(aes(sample = log(maizC1))) +
  geom_abline(intercept = mean(log(datosRindeMaiz@data$maizC1), na.rm = TRUE),
              slope = sd(log(datosRindeMaiz@data$maizC1), na.rm = TRUE), col = "red") +
  labs(title = "Log Rend.maíz C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC1Sqrt <- ggDatosRindeMaiz +
  geom_qq(aes(sample = sqrt(maizC1))) +
  geom_abline(intercept = mean(sqrt(datosRindeMaiz@data$maizC1), na.rm = TRUE),
              slope = sd(sqrt(datosRindeMaiz@data$maizC1), na.rm = TRUE), col = "red") +
  labs(title = "Raíz cuadrada Rend.maíz C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC1BC <- ggDatosRindeMaiz +
  geom_qq(aes(sample = maizC1BC)) +
  geom_abline(intercept = mean(maizC1BC, na.rm = TRUE),
              slope = sd(maizC1BC, na.rm = TRUE), col = "red") +
  labs(title = "Box-Cox Rend.maíz C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

# Q-Q plot rendimiento maíz (C2)
ggQQmaizC2 <- ggDatosRindeMaiz +
  geom_qq(aes(sample = maizC2)) +
  geom_abline(intercept = mean(datosRindeMaiz@data$maizC2, na.rm = TRUE),
              slope = sd(datosRindeMaiz@data$maizC2, na.rm = TRUE), col = "red") +
  labs(title = "Rend.maíz C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC2Log <- ggDatosRindeMaiz +
  geom_qq(aes(sample = log(maizC2))) +
  geom_abline(intercept = mean(log(datosRindeMaiz@data$maizC2), na.rm = TRUE),
              slope = sd(log(datosRindeMaiz@data$maizC2), na.rm = TRUE), col = "red") +
  labs(title = "Log Rend.maíz C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC2Sqrt <- ggDatosRindeMaiz +
  geom_qq(aes(sample = sqrt(maizC2))) +
  geom_abline(intercept = mean(sqrt(datosRindeMaiz@data$maizC2), na.rm = TRUE),
```

```

            slope = sd(sqrt(datosRindeMaiz@data$maizC2), na.rm = TRUE), col = "red") +
  labs(title = "Raíz cuadrada Rend.maíz C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQmaizC2BC <- ggDatosRindeMaiz +
  geom_qq(aes(sample = maizC2BC)) +
  geom_abline(intercept = mean(maizC2BC, na.rm = TRUE),
              slope = sd(maizC2BC, na.rm = TRUE), col = "red") +
  labs(title = "Box-Cox Rend.maíz C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

# Q-Q plot profundidad de la napa (C1)
ggQQNapaDicC1 <- ggDatosRindeMaiz +
  geom_qq(aes(sample = NapaDicC1)) +
  geom_abline(intercept = mean(datosRindeMaiz@data$NapaDicC1),
              slope = sd(datosRindeMaiz@data$NapaDicC1), col = "red") +
  labs(title = "Prof.napa C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQNapaDicC1Log <- ggDatosRindeMaiz +
  geom_qq(aes(sample = log(NapaDicC1))) +
  geom_abline(intercept = mean(log(datosRindeMaiz@data$NapaDicC1)),
              slope = sd(log(datosRindeMaiz@data$NapaDicC1)), col = "red") +
  labs(title = "Log Prof.napa C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQNapaDicC1Sqrt <- ggDatosRindeMaiz +
  geom_qq(aes(sample = sqrt(NapaDicC1))) +
  geom_abline(intercept = mean(sqrt(datosRindeMaiz@data$NapaDicC1)),
              slope = sd(sqrt(datosRindeMaiz@data$NapaDicC1)), col = "red") +
  labs(title = "Raíz cuadrada Prof.napa C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQNapaDicC1BC <- ggDatosRindeMaiz +
  geom_qq(aes(sample = NapaDicC1BC)) +
  geom_abline(intercept = mean(NapaDicC1BC, na.rm = TRUE),
              slope = sd(NapaDicC1BC, na.rm = TRUE), col = "red") +
  labs(title = "Box-Cox Prof.napa C1", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

# Q-Q plot profundidad de la napa (C2)
ggQQNapaDicC2 <- ggDatosRindeMaiz +
  geom_qq(aes(sample = NapaDicC2)) +
  geom_abline(intercept = mean(datosRindeMaiz@data$NapaDicC2),
              slope = sd(datosRindeMaiz@data$NapaDicC2), col = "red") +
  labs(title = "Prof.napa C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQNapaDicC2Log <- ggDatosRindeMaiz +
  geom_qq(aes(sample = log(NapaDicC2))) +
  geom_abline(intercept = mean(log(datosRindeMaiz@data$NapaDicC2)),
              slope = sd(log(datosRindeMaiz@data$NapaDicC2)), col = "red") +
  labs(title = "Log Prof.napa C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

```

```

y = "Cuantiles de la muestra")

ggQQNapaDicC2Sqrt <- ggDatosRindeMaiz +
  geom_qq(aes(sample = sqrt(NapaDicC2))) +
  geom_abline(intercept = mean(sqrt(datosRindeMaiz@data$NapaDicC2)),
              slope = sd(sqrt(datosRindeMaiz@data$NapaDicC2)), col = "red") +
  labs(title = "Raíz cuadrada Prof.napa C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQNapaDicC2BC <- ggDatosRindeMaiz +
  geom_qq(aes(sample = NapaDicC2BC)) +
  geom_abline(intercept = mean(NapaDicC2BC, na.rm = TRUE),
              slope = sd(NapaDicC2BC, na.rm = TRUE), col = "red") +
  labs(title = "Box-Cox Prof.napa C2", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

# Q-Q plot elevación
ggQQElev <- ggDatosRindeMaiz +
  geom_qq(aes(sample = Elev)) +
  geom_abline(intercept = mean(datosRindeMaiz@data$Elev),
              slope = sd(datosRindeMaiz@data$Elev), col = "red") +
  labs(title = "Elevación", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

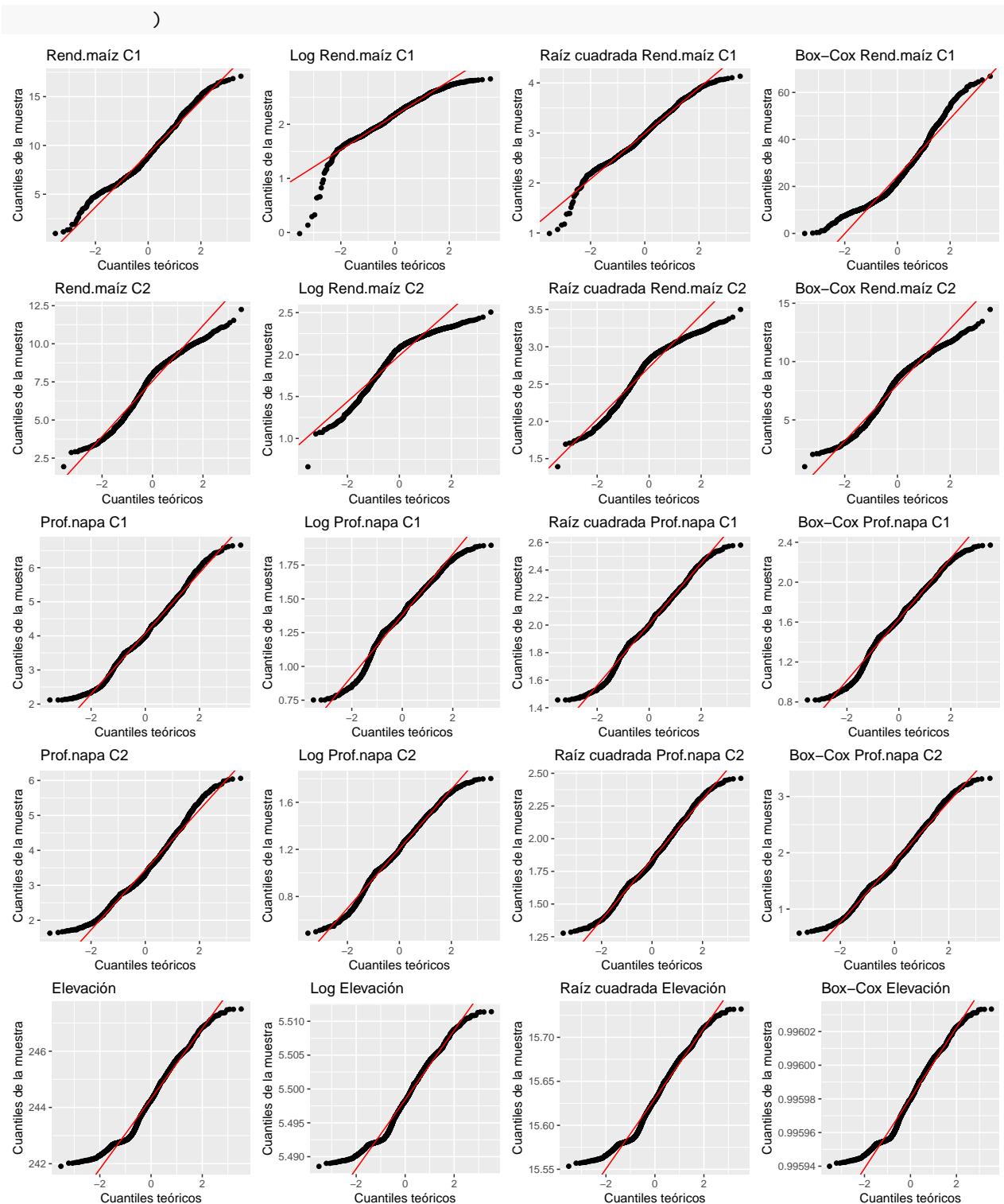
ggQQElevLog <- ggDatosRindeMaiz +
  geom_qq(aes(sample = log(Elev))) +
  geom_abline(intercept = mean(log(datosRindeMaiz@data$Elev)),
              slope = sd(log(datosRindeMaiz@data$Elev)), col = "red") +
  labs(title = "Log Elevación", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQElevSqrt <- ggDatosRindeMaiz +
  geom_qq(aes(sample = sqrt(Elev))) +
  geom_abline(intercept = mean(sqrt(datosRindeMaiz@data$Elev)),
              slope = sd(sqrt(datosRindeMaiz@data$Elev)), col = "red") +
  labs(title = "Raíz cuadrada Elevación", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

ggQQElevBC <- ggDatosRindeMaiz +
  geom_qq(aes(sample = ElevBC)) +
  geom_abline(intercept = mean(ElevBC), slope = sd(ElevBC), col = "red") +
  labs(title = "Box-Cox Elevación", x = "Cuantiles teóricos",
       y = "Cuantiles de la muestra")

# Ver plots
grid.arrange(ggQQmaizC1, ggQQmaizC2, ggQQNapaDicC1, ggQQNapaDicC2, ggQQElev,
             ggQQmaizC1Log, ggQQmaizC2Log, ggQQNapaDicC1Log, ggQQNapaDicC2Log,
             ggQQElevLog, ggQQmaizC1Sqrt, ggQQmaizC2Sqrt, ggQQNapaDicC1Sqrt,
             ggQQNapaDicC2Sqrt, ggQQElevSqrt, ggQQmaizC1BC, ggQQmaizC2BC,
             ggQQNapaDicC1BC, ggQQNapaDicC2BC, ggQQElevBC,
             ncol = 4, nrow = 5,
             layout_matrix = cbind(c(1,2,3,4,5), c(6,7,8,9,10),
                                   c(11,12,13,14,15), c(16,17,18,19,20)))

```



A partir del análisis de los Q-Q plots (cuantiles - cuantiles) se observó que todas las transformaciones probadas no acercaron sustancialmente los datos originales a distribuciones normales. Incluso, los Q-Q plots para los datos originales confirmaron el desvío de los datos de una distribución normal.

Análisis de tendencias

Para analizar si existen tendencias entre el *rendimiento de maíz* y las covariables ambientales se realizaron gráficos de dispersión de puntos junto con líneas basadas en suavizados de la medias locales de los valores.

```
# Rendimiento maíz vs Longitud
ggTend_MaizC1VsLong <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC1, x = datosRindeMaiz@coords[, "Long"])) +
  geom_smooth(aes(y = maizC1, x = datosRindeMaiz@coords[, "Long"])) +
  labs(title = "", x = "Longitud", y = "Rendimiento maíz C1 (t/ha)")

ggTend_MaizC2VsLong <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC2, x = datosRindeMaiz@coords[, "Long"])) +
  geom_smooth(aes(y = maizC2, x = datosRindeMaiz@coords[, "Long"])) +
  labs(title = "", x = "Longitud", y = "Rendimiento maíz C2 (t/ha)")

# Rendimiento maíz vs Latitud
ggTend_MaizC1VsLat <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC1, x = datosRindeMaiz@coords[, "Lat"])) +
  geom_smooth(aes(y = maizC1, x = datosRindeMaiz@coords[, "Lat"])) +
  labs(title = "", x = "Latitud", y = "Rendimiento maíz C1 (t/ha)")

ggTend_MaizC2VsLat <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC2, x = datosRindeMaiz@coords[, "Lat"])) +
  geom_smooth(aes(y = maizC2, x = datosRindeMaiz@coords[, "Lat"])) +
  labs(title = "", x = "Latitud", y = "Rendimiento maíz C2 (t/ha)")

# Rendimiento maíz vs Elevación
ggTend_MaizC1VsElev <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC1, x = Elev)) +
  geom_smooth(aes(y = maizC1, x = Elev)) +
  labs(title = "", y = "Rendimiento maíz C1 (t/ha)",
       x = "Elevación (m)")

ggTend_MaizC2VsElev <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC2, x = Elev)) +
  geom_smooth(aes(y = maizC2, x = Elev)) +
  labs(title = "", y = "Rendimiento maíz C2 (t/ha)",
       x = "Elevación (m)")

# Rendimiento maíz vs Profundidad de la napa
ggTend_MaizC1VsNapaDicC1 <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC1, x = NapaDicC1)) +
  geom_smooth(aes(y = maizC1, x = NapaDicC1)) +
  labs(title = "", y = "Rendimiento maíz C1 (t/ha)",
       x = "Profundidad napa (m)")

ggTend_MaizC2VsNapaDicC2 <- ggDatosRindeMaiz +
  geom_point(aes(y = maizC2, x = NapaDicC2)) +
  geom_smooth(aes(y = maizC2, x = NapaDicC2)) +
  labs(title = "", y = "Rendimiento maíz C2 (t/ha)",
       x = "Profundidad napa (m)")

# Profundidad de la napa vs Elevación
```

```

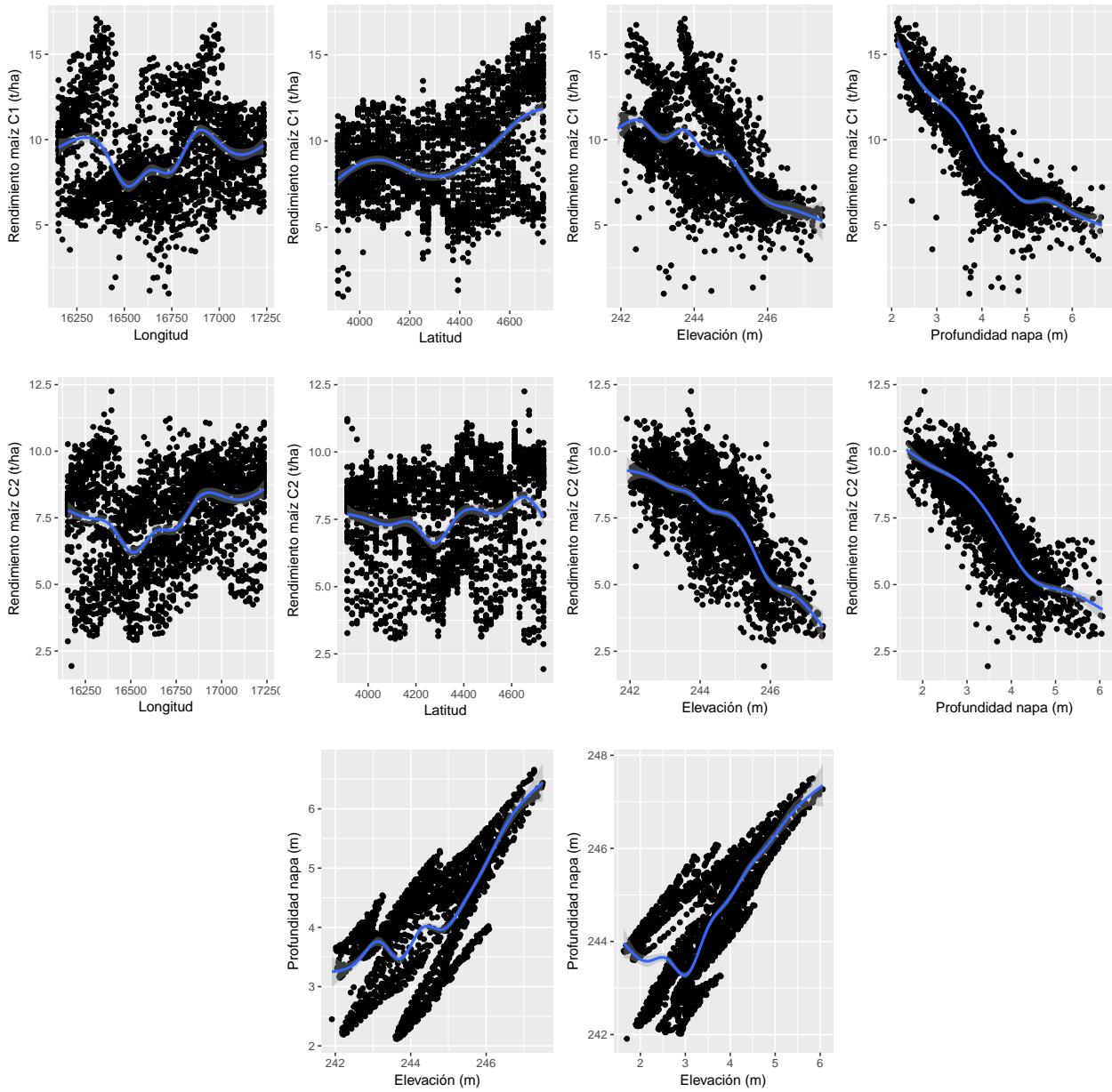
ggTend_NapaDicC1VsElev <- ggDatosRindeMaiz +
  geom_point(aes(x = Elev, y = NapaDicC1)) +
  geom_smooth(aes(x = Elev, y = NapaDicC1)) +
  labs(title = "", x = "Elevación (m)", y = "Profundidad napa (m)")

ggTend_NapaDicC2VsElev <- ggDatosRindeMaiz +
  geom_point(aes(y = Elev, x = NapaDicC2)) +
  geom_smooth(aes(y = Elev, x = NapaDicC2)) +
  labs(title = "", x = "Elevación (m)", y = "Profundidad napa (m)")

# Ver plots

grid.arrange(ggTend_MaizC1VsLong, ggTend_MaizC1VsLat,
             ggTend_MaizC1VsElev, ggTend_MaizC1VsNapaDicC1,
             ggTend_MaizC2VsLong, ggTend_MaizC2VsLat,
             ggTend_MaizC2VsElev, ggTend_MaizC2VsNapaDicC2,
             ggTend_NapaDicC1VsElev, ggTend_NapaDicC2VsElev,
             ncol = 4, nrow = 3,
             layout_matrix = rbind(c(1,2,3,4), c(5,6,7,8), c(NA,9,10,NA)))
)

```



A partir de los gráficos de dispersión y suavizado de medias, no se observó una tendencia entre el *rendimiento de maíz* y la *latitud* o *longitud*. Sin embargo, se observaron tendencias negativas entre el *rendimiento de maíz* y la *elevación* y también con la *profundidad de la napa* para las dos campañas. Se observó que el *rendimiento de maíz* decrece con la *elevación* del terreno y la *profundidad de la napa*. Asimismo, se observó una tendencia positiva entre la *profundidad de la napa* y la *elevación*.

Correlación

Se calculó el coeficiente de correlación de Pearson (r) para cuantificar la relación entre las variables para las cuales se observó tendencia y para las que no. Para complementar la interpretación, se realizaron correlogramas para visualizar la matriz de correlaciones.

```
# Rendimiento maíz vs Longitud para C1
cor.test(x = datosRindeMaiz@coords[-NAs, "Long"],
```

```

y = datosRindeMaiz@data$maizC1[-NAs],
method = "pearson")

## Pearson's product-moment correlation
## data: datosRindeMaiz@coords[-NAs, "Long"] and datosRindeMaiz@data$maizC1[-NAs]
## t = 1.5344, df = 2306, p-value = 0.1251
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.008876447 0.072642586
## sample estimates:
## cor
## 0.03193618

# Rendimiento maíz vs Longitud para C2
cor.test(x = datosRindeMaiz@coords[, "Long"],
          y = datosRindeMaiz@data$maizC2,
          method = "pearson")

## Pearson's product-moment correlation
## data: datosRindeMaiz@coords[, "Long"] and datosRindeMaiz@data$maizC2
## t = 11.793, df = 2308, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.1995606 0.2764987
## sample estimates:
## cor
## 0.2384037

# Rendimiento maíz vs Latitud para C1
cor.test(x = datosRindeMaiz@coords[-NAs, "Lat"],
          y = datosRindeMaiz@data$maizC1[-NAs],
          method = "pearson")

## Pearson's product-moment correlation
## data: datosRindeMaiz@coords[-NAs, "Lat"] and datosRindeMaiz@data$maizC1[-NAs]
## t = 17.82, df = 2306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.3115241 0.3832638
## sample estimates:
## cor
## 0.3479031

# Rendimiento maíz vs Latitud para C2
cor.test(x = datosRindeMaiz@coords[, "Lat"],
          y = datosRindeMaiz@data$maizC2,
          method = "pearson")

## Pearson's product-moment correlation

```

```

## 
## data: datosRindeMaiz@coords[, "Lat"] and datosRindeMaiz@data$maizC2
## t = 5.8751, df = 2308, p-value = 4.836e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08100608 0.16137293
## sample estimates:
##      cor
## 0.1213884

# Rendimiento maíz vs Elevación para C1
cor.test(x = datosRindeMaiz@data$Elev[-NAs],
          y = datosRindeMaiz@data$maizC1[-NAs],
          method = "pearson")

## 
## Pearson's product-moment correlation
## 
## data: datosRindeMaiz@data$Elev[-NAs] and datosRindeMaiz@data$maizC1[-NAs]
## t = -32.254, df = 2306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.5850616 -0.5287992
## sample estimates:
##      cor
## -0.5575704

# Rendimiento maíz vs Elevación para C2
cor.test(x = datosRindeMaiz@data$Elev,
          y = datosRindeMaiz@data$maizC2,
          method = "pearson")

## 
## Pearson's product-moment correlation
## 
## data: datosRindeMaiz@data$Elev and datosRindeMaiz@data$maizC2
## t = -51.115, df = 2308, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.7472524 -0.7089611
## sample estimates:
##      cor
## -0.7286757

# Rendimiento maíz vs Profundidad de la napa para C1
cor.test(x = datosRindeMaiz@data$NapaDicC1[-NAs],
          y = datosRindeMaiz@data$maizC1[-NAs],
          method = "pearson")

## 
## Pearson's product-moment correlation
## 
## data: datosRindeMaiz@data$NapaDicC1[-NAs] and datosRindeMaiz@data$maizC1[-NAs]
## t = -80.658, df = 2306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:

```

```

## -0.8695615 -0.8481802
## sample estimates:
## cor
## -0.8592456
# Rendimiento maíz vs Profundidad de la napa para C2
cor.test(x = datosRindeMaiz@data$NapaDicC2,
          y = datosRindeMaiz@data$maizC2,
          method = "pearson")

##
## Pearson's product-moment correlation
##
## data: datosRindeMaiz@data$NapaDicC2 and datosRindeMaiz@data$maizC2
## t = -74.21, df = 2308, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.8510958 -0.8269790
## sample estimates:
## cor
## -0.8394502

# Profundidad de la napa vs Elevación para C1
cor.test(x = datosRindeMaiz@data$NapaDicC1[-NAs],
          y = datosRindeMaiz@data$Elev[-NAs],
          method = "pearson")

##
## Pearson's product-moment correlation
##
## data: datosRindeMaiz@data$NapaDicC1[-NAs] and datosRindeMaiz@data$Elev[-NAs]
## t = 43.721, df = 2306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.6502840 0.6949353
## sample estimates:
## cor
## 0.6732229

# Profundidad de la napa vs Elevación para C2
cor.test(x = datosRindeMaiz@data$NapaDicC2,
          y = datosRindeMaiz@data$Elev,
          method = "pearson")

##
## Pearson's product-moment correlation
##
## data: datosRindeMaiz@data$NapaDicC2 and datosRindeMaiz@data$Elev
## t = 53.189, df = 2308, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.7232072 0.7598875
## sample estimates:
## cor
## 0.7421024

```

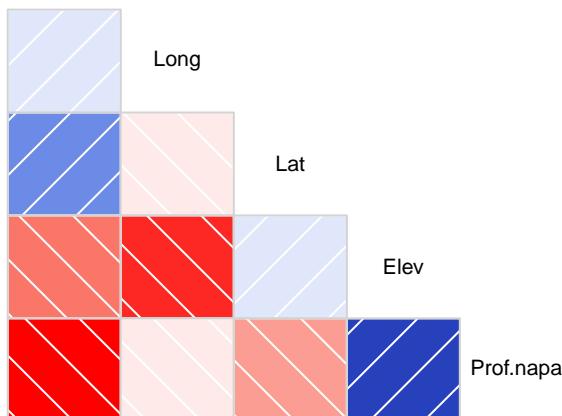
En los siguientes correlogramas las celdas de colores azules y con un patrón de relleno de líneas con una

pendiente positiva representan correlaciones positivas y las celdas de colores rojos con un patrón de relleno de líneas con una pendiente negativa representan correlaciones negativas. La intensidad del color es proporcional a la magnitud del coeficiente de correlación de Pearson.

```
# Correlograma C1
corrgram(data.frame("Rend.maíz" = datosRindeMaiz@data$maizC1,
                    "Long" = datosRindeMaiz@coords[, "Long"],
                    "Lat" = datosRindeMaiz@coords[, "Lat"],
                    "Elev" = datosRindeMaiz@data$Elev,
                    "Prof.napa" = datosRindeMaiz@data$NapaDicC1),
          cor.method="pearson",
          order = NULL,
          lower.panel = panel.shade,
          upper.panel = NULL,
          text.panel = panel.txt,
          main = "Correlograma - C1")
```

Correlograma – C1

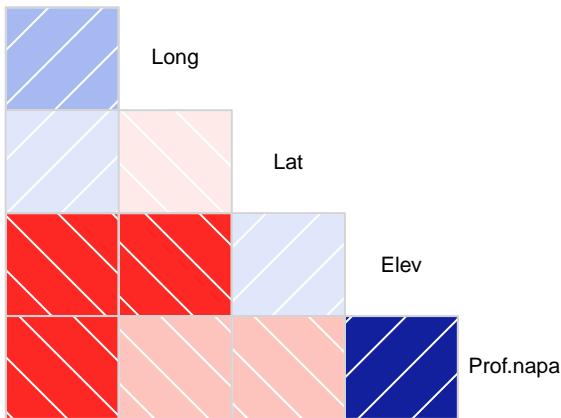
Rend.maíz



```
# Correlograma C2
corrgram(data.frame("Rend.maíz" = datosRindeMaiz@data$maizC2,
                    "Long" = datosRindeMaiz@coords[, "Long"],
                    "Lat" = datosRindeMaiz@coords[, "Lat"],
                    "Elev" = datosRindeMaiz@data$Elev,
                    "Prof.napa" = datosRindeMaiz@data$NapaDicC2),
          cor.method="pearson",
          order = NULL,
          lower.panel = panel.shade,
          upper.panel = NULL,
          text.panel = panel.txt,
          main = "Correlograma - C2")
```

Correlograma – C2

Rend.maíz



Se observó que la correlación entre el *rendimiento de maíz* con la *latitud* y la *longitud* fueron bajas ($0 < r < 0.35$), confirmando una tendencia despreciable. Por otro lado, se observó que la correlación fue alta para el *rendimiento de maíz* con la *elevación* ($r_{C1} = -0.56$ y $r_{C2} = -0.73$) y aún fue mayor para la *profundidad de la napa* ($r_{C1} = -0.86$ y $r_{C2} = -0.84$). Se confirmó que existe una tendencia negativa entre el *rendimiento de maíz* y la *elevación* y entre el *rendimiento de maíz* y la *profundidad de la napa*. También se observó que existe una correlación positiva entre la *profundidad de la napa* y la *elevación* ($r_{C1} = 0.67$ y $r_{C2} = 0.74$).

Para definir el modelo de tendencia del *rendimiento de maíz*, se evaluaron tres modelos de regresiones lineales:

1. Modelo de tendencia basado en una regresión lineal múltiple del *rendimiento de maíz* con la *profundidad de la napa* y la *elevación*.
2. Modelo de tendencia basado en una regresión lineal simple del *rendimiento de maíz* con la *profundidad de la napa*.
3. Modelo de tendencia basado en una regresión lineal simple del *rendimiento de maíz* con la *elevación*.

Se utilizó el criterio de información de Akaike (*AIC*) para definir el modelo (a menor *AIC* mejor ajuste).

```
# Ajuste de modelos para C1
ajusteC1.M1 <- lm(formula = maizC1 ~ NapaDicC1 + Elev, data = datosRindeMaiz)
ajusteC1.M2 <- lm(formula = maizC1 ~ NapaDicC1, data = datosRindeMaiz)
ajusteC1.M3 <- lm(formula = maizC1 ~ Elev, data = datosRindeMaiz)
```

```
# Criterio de Información de Akaike
AIC(ajusteC1.M1, ajusteC1.M2, ajusteC1.M3)
```

```
##          df      AIC
## ajusteC1.M1  4  8085.556
## ajusteC1.M2  3  8090.608
## ajusteC1.M3  3 10325.262
```

```
# Ajuste de moelos para C2
ajusteC2.M1 <- lm(formula = maizC2 ~ NapaDicC2 + Elev, data = datosRindeMaiz)
ajusteC2.M2 <- lm(formula = maizC2 ~ NapaDicC2, data = datosRindeMaiz)
ajusteC2.M3 <- lm(formula = maizC2 ~ Elev, data = datosRindeMaiz)
```

```
# Criterio de Información de Akaike
```

```
AIC(ajusteC2.M1, ajusteC2.M2, ajusteC2.M3)
```

```
##          df      AIC
## ajusteC2.M1  4 6306.514
## ajusteC2.M2  3 6507.776
## ajusteC2.M3  3 7576.382
```

El mejor modelo de tendencia para las dos campañas del *rendimiento de maíz* según el criterio de información de Akaike fue el de regresión múltiple ($y = \beta_1x_1 + \beta_2x_1 + \beta_0$). Además, se calcularon métricas para conocer la importancia relativa de cada una de las variables en el modelo.

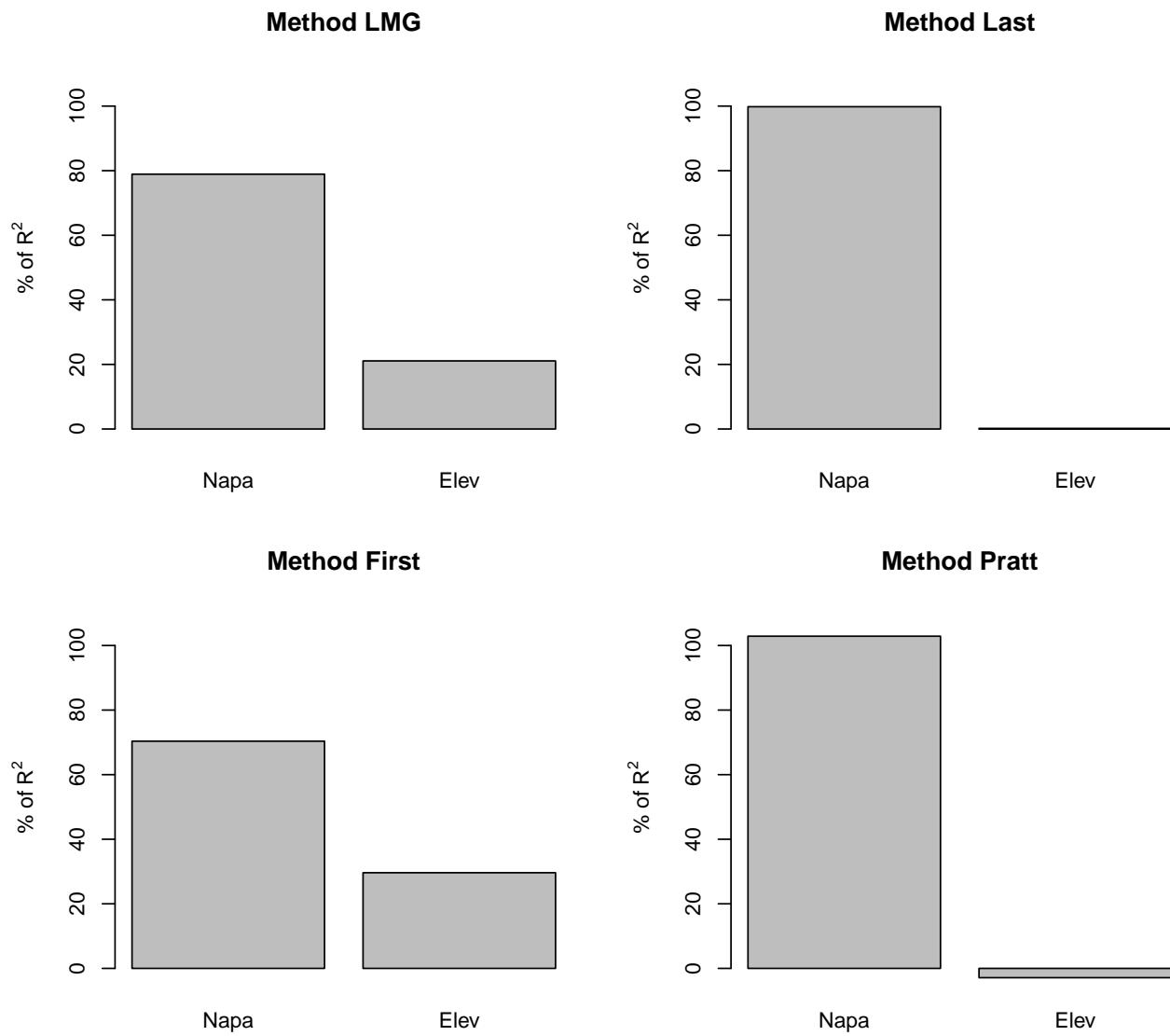
```
# Calcular la importancia relativa para cada predictor - C1
```

```
impRel.C1.M1 <- calc.relimp(ajusteC1.M1,
                             type = c("lmg", "last", "first", "pratt"),
                             rela = TRUE)
# Resumen
```

```
(impRel.C1.M1)
```

```
## Response variable: maizC1
## Total response variance: 7.433428
## Analysis based on 2308 observations
##
## 2 Regressors:
## NapaDicC1 Elev
## Proportion of variance explained by model: 73.91%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##           lmg      last     first     pratt
## NapaDicC1 0.7891472 0.99813902 0.7036901  1.02882704
## Elev       0.2108528 0.00186098 0.2963099 -0.02882704
##
## Average coefficients for different model sizes:
##
##           1X      2Xs
## NapaDicC1 -2.642097 -2.72120003
## Elev       -1.216533  0.08337362
# Gráfico
plot(impRel.C1.M1)
```

Relative importances for maizC1



$R^2 = 73.91\%$, metrics are normalized to sum 100%.

```
# Calcular la importancia relativa para cada predictor - C2
impRel.C2.M1 <- calc.relimp(ajusteC2.M1,
                           type = c("lmg", "last", "first", "pratt"),
                           rela = TRUE)

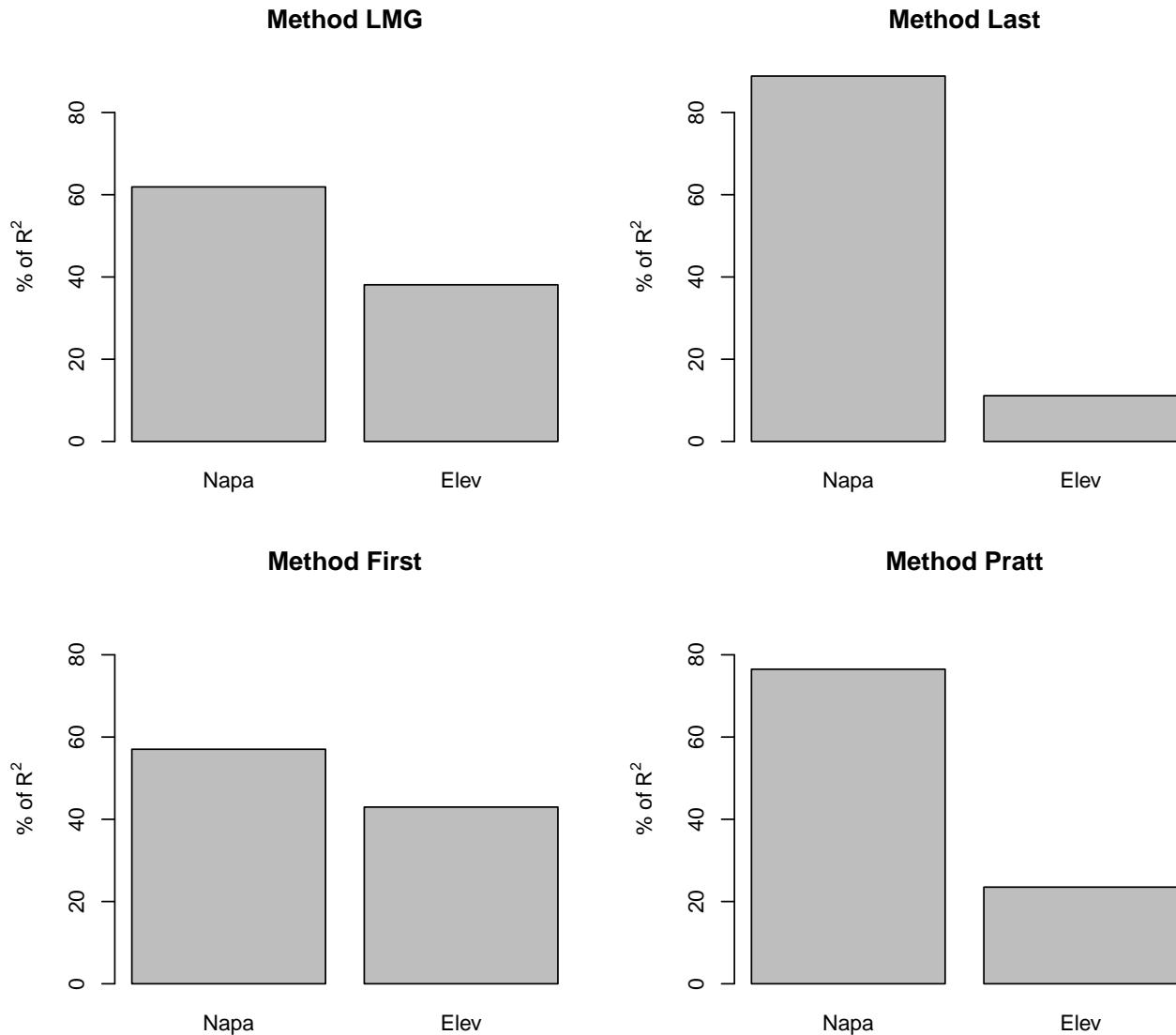
# Resumen
(impRel.C2.M1 )
```

```
## Response variable: maizC2
## Total response variance: 3.309716
## Analysis based on 2310 observations
##
## 2 Regressors:
## NapaDicC2 Elev
```

```
## Proportion of variance explained by model: 72.96%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##          lmg      last     first     pratt
## NapaDicC2 0.6190514 0.8886796 0.5702906 0.7649802
## Elev       0.3809486 0.1113204 0.4297094 0.2350198
##
## Average coefficients for different model sizes:
##
##          1X      2Xs
## NapaDicC2 -1.778489 -1.4085359
## Elev       -1.061044 -0.3426301

# Gráfico
plot(impRel.C2.M1)
```

Relative importances for maizC2



$R^2 = 72.96\%$, metrics are normalized to sum 100%.

La varianza explicada por el modelo elegido fue de 73.91% y de 72.96% para C1 y para C2 respectivamente. Se observó que la *profundidad de la napa* tuvo una importancia relativa mayor en el modelo de tendencia elegido que la *elevación*. Además, dicha observación fue más acentuada para la C1 que para la C2.

```
# Agregar modelo de tendencia y residuales a los datos
```

```
# C1
datosRindeMaiz$ajusteC1.M1 <-
  ajusteC1.M1$coefficients["NapaDicC1"] * datosRindeMaiz$NapaDicC1 +
  ajusteC1.M1$coefficients["Elev"] * datosRindeMaiz$Elev +
  ajusteC1.M1$coefficients["(Intercept)"]

datosRindeMaiz$resAjusteC1.M1 <- NA
datosRindeMaiz$resAjusteC1.M1[-NAs] <- residuals(ajusteC1.M1)
```

```

# C2
datosRindeMaiz$ajusteC2.M1 <-
  ajusteC2.M1$coefficients["NapaDicC2"] * datosRindeMaiz$NapaDicC2 +
  ajusteC2.M1$coefficients["Elev"] * datosRindeMaiz$Elev +
  ajusteC2.M1$coefficients["(Intercept)"]

datosRindeMaiz$resAjusteC2.M1 <- residuals(ajusteC2.M1)

```

Autocorrelación espacial

Para examinar patrones de autocorrelación espacial se realizaron correlogramas espaciales basados en el índice de Moran (I) para los residuales del modelo de tendencia elegido del *rendimiento de maíz*. El color azul muestra autocorrelación espacial positiva mientras que el color rojo muestra autocorrelación espacial negativa.

```

# Índice de Moran para el rendimiento de maíz - C1
moranRendMaizC1 <- correlog(x = as.data.frame(datosRindeMaiz@coords)$Long,
                               y = as.data.frame(datosRindeMaiz@coords)$Lat,
                               z = datosRindeMaiz$resAjusteC1.M1,
                               increment = 100,
                               resamp = 0,
                               latlon = FALSE,
                               na.rm = TRUE)

# Índice de Moran para el rendimiento de maíz - C2
moranRendMaizC2 <- correlog(x = as.data.frame(datosRindeMaiz@coords)$Long,
                               y = as.data.frame(datosRindeMaiz@coords)$Lat,
                               z = datosRindeMaiz$resAjusteC2.M1,
                               increment = 100,
                               resamp = 0,
                               latlon = FALSE,
                               na.rm = TRUE)

```

En los correlogramas se observó una baja correlación espacial positiva hasta los $150m - 200m$. A partir de esa distancia la correlación espacial fue aleatoria (valores entorno a cero). Para la C2 se observó un valor muy alto correlación positiva para distancias mayores a $1200m$ que pudo deberse a la presencia de valores atípicos y/o a un n bajo para esa clase de distancia ($n_{14} = 96$).

```

# Plot correlograma

# Polígonos de tipos de correlación
polyAutocorrPos <- data.frame(
  "id" = rep("1", length(moranRendMaizC2$mean.of.class)/2),
  "x" = c(seq(0, max(moranRendMaizC2$mean.of.class),
            length.out = length(moranRendMaizC2$mean.of.class)/2),
          rev(seq(0, max(moranRendMaizC2$mean.of.class),
                 length.out = length(moranRendMaizC2$mean.of.class)/2))),
  "y" = c(rep(0,7), rep(1,7)))

polyAutocorrNeg <- data.frame(
  "id" = rep("2", length(moranRendMaizC2$mean.of.class)/2),
  "x" = c(seq(0, max(moranRendMaizC2$mean.of.class),
            length.out = length(moranRendMaizC2$mean.of.class)/2),
          rev(seq(0, max(moranRendMaizC2$mean.of.class),
                 length.out = length(moranRendMaizC2$mean.of.class)/2)))

```

```

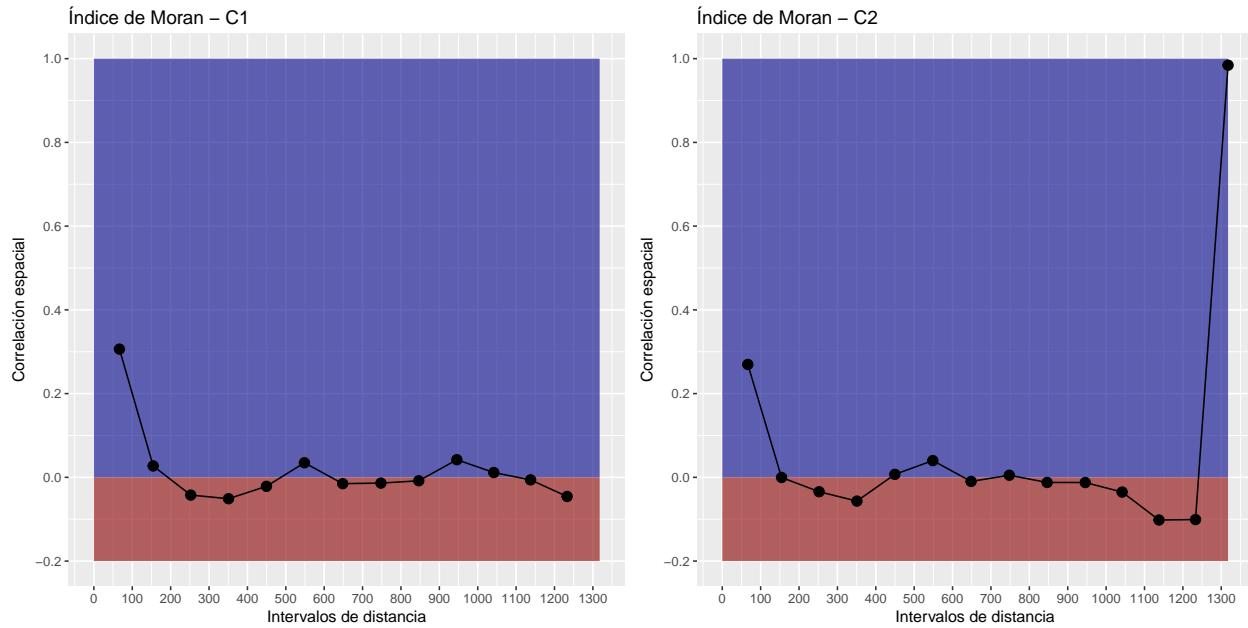
    rev(seq(0, max(moranRendMaizC2$mean.of.class),
            length.out = length(moranRendMaizC2$mean.of.class)/2))),
    "y" = c(rep(0,7), rep(-0.2,7)))

# Índice de Moran para el rendimiento de maíz - C1
ggMoranRendMaizC1 <- ggplot(data.frame("intervalos" = moranRendMaizC1$mean.of.class,
                                         "correlación" = moranRendMaizC1$correlation)) +
  geom_polygon(aes(x = polyAutocorrPos$x, y = polyAutocorrPos$y,
                    group = polyAutocorrPos$id),
               fill = "darkblue", alpha = 0.6) +
  geom_polygon(aes(x = polyAutocorrNeg$x, y = polyAutocorrNeg$y,
                    group = polyAutocorrNeg$id),
               fill = "darkred", alpha = 0.6) +
  geom_point(aes(x = intervalos, y = correlación), size = 3) +
  geom_line(aes(x = intervalos, y = correlación)) +
  labs(title = "Índice de Moran - C1",
       x = "Intervalos de distancia",
       y = "Correlación espacial") +
  scale_x_continuous(limits = c(0, max(moranRendMaizC1$mean.of.class)),
                     breaks = seq(0, max(moranRendMaizC1$mean.of.class), 100)) +
  scale_y_continuous(limits = c(-0.2, 1), breaks = seq(-0.2, 1, 0.2))

# Plot correlograma
# Índice de Moran para el rendimiento de maíz - C2
ggMoranRendMaizC2 <- ggplot(data.frame("intervalos" = moranRendMaizC2$mean.of.class,
                                         "correlación" = moranRendMaizC2$correlation)) +
  geom_polygon(aes(x = polyAutocorrPos$x, y = polyAutocorrPos$y,
                    group = polyAutocorrPos$id),
               fill = "darkblue", alpha = 0.6) +
  geom_polygon(aes(x = polyAutocorrNeg$x, y = polyAutocorrNeg$y,
                    group = polyAutocorrNeg$id),
               fill = "darkred", alpha = 0.6) +
  geom_point(aes(x = intervalos, y = correlación), size = 3) +
  geom_line(aes(x = intervalos, y = correlación)) +
  labs(title = "Índice de Moran - C2",
       x = "Intervalos de distancia",
       y = "Correlación espacial") +
  scale_x_continuous(limits = c(0, max(moranRendMaizC2$mean.of.class)),
                     breaks = seq(0, max(moranRendMaizC2$mean.of.class), 100)) +
  scale_y_continuous(limits = c(-0.2, 1), breaks = seq(-0.2, 1, 0.2))

# Ver plot
grid.arrange(ggMoranRendMaizC1, ggMoranRendMaizC2,
             ncol = 2, nrow = 1,
             layout_matrix = cbind(1,2)
           )

```

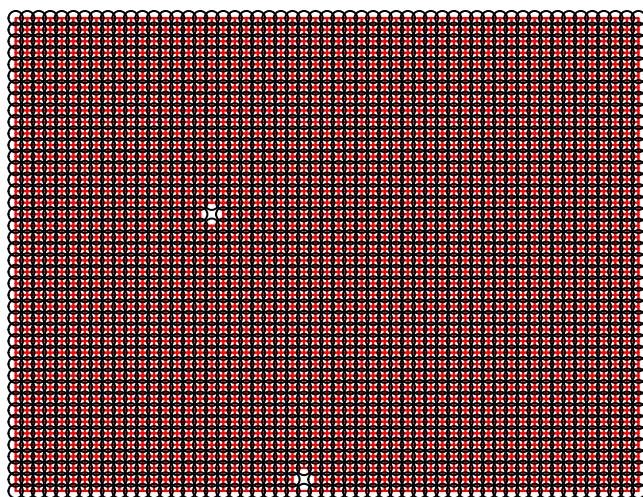


Identificación de valores atípicos espaciales

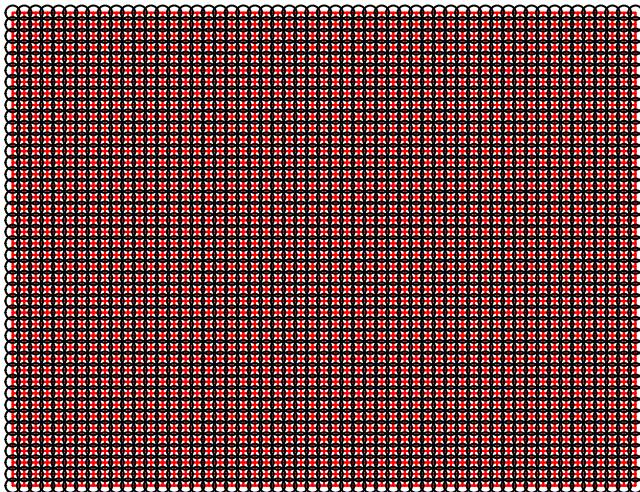
Para detectar valores atípicos espaciales se analizó el gráfico de dispersión de Moran que permite evaluar que tan similar es un valor observado respecto a las observaciones de sus vecinos en el espacio. Los puntos en los cuadrantes superior derecho e inferior izquierdo indican una asociación espacial positiva de valores que son más altos y más bajos que la media muestral respectivamente. Los puntos de los cuadrantes inferior derecho y superior izquierdo incluyen observaciones que indican una asociación espacial negativa, es decir, poseen poca similitud con sus vecinos.

```
# Crear matriz de vecinos por distancias
nbC1 <- dnearest(datosRindeMaiz[-NAs,], 0, 25)
nbC2 <- dnearest(datosRindeMaiz, 0, 25)

# Plotear matriz espacial de vecinos
plot.nb(x = nbC1, coords = datosRindeMaiz@coords[-NAs,], col = "red")
```



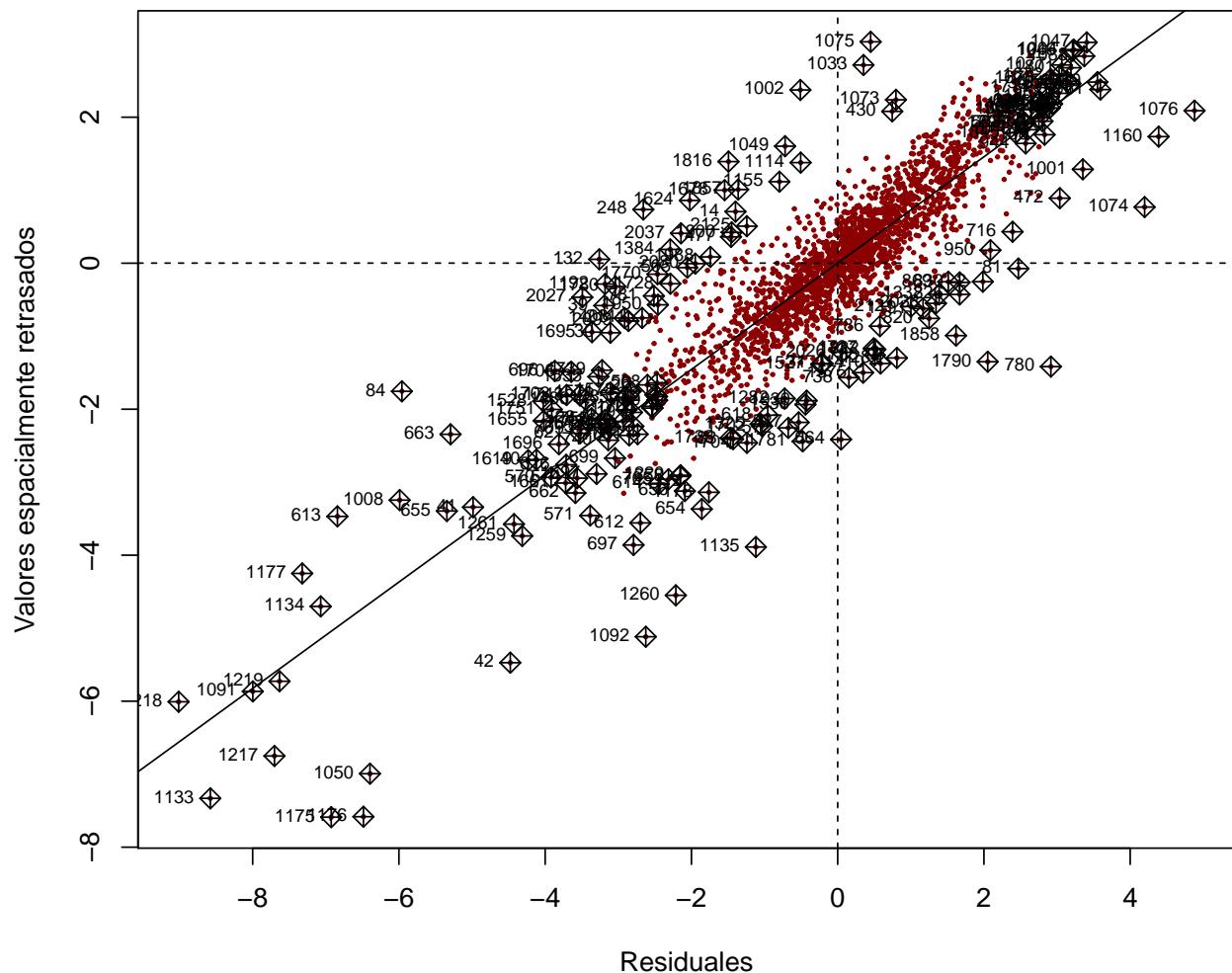
```
plot.nb(x = nbC2, coords = datosRindeMaiz@coords, col = "red")
```



Plot de dispersión de Moran

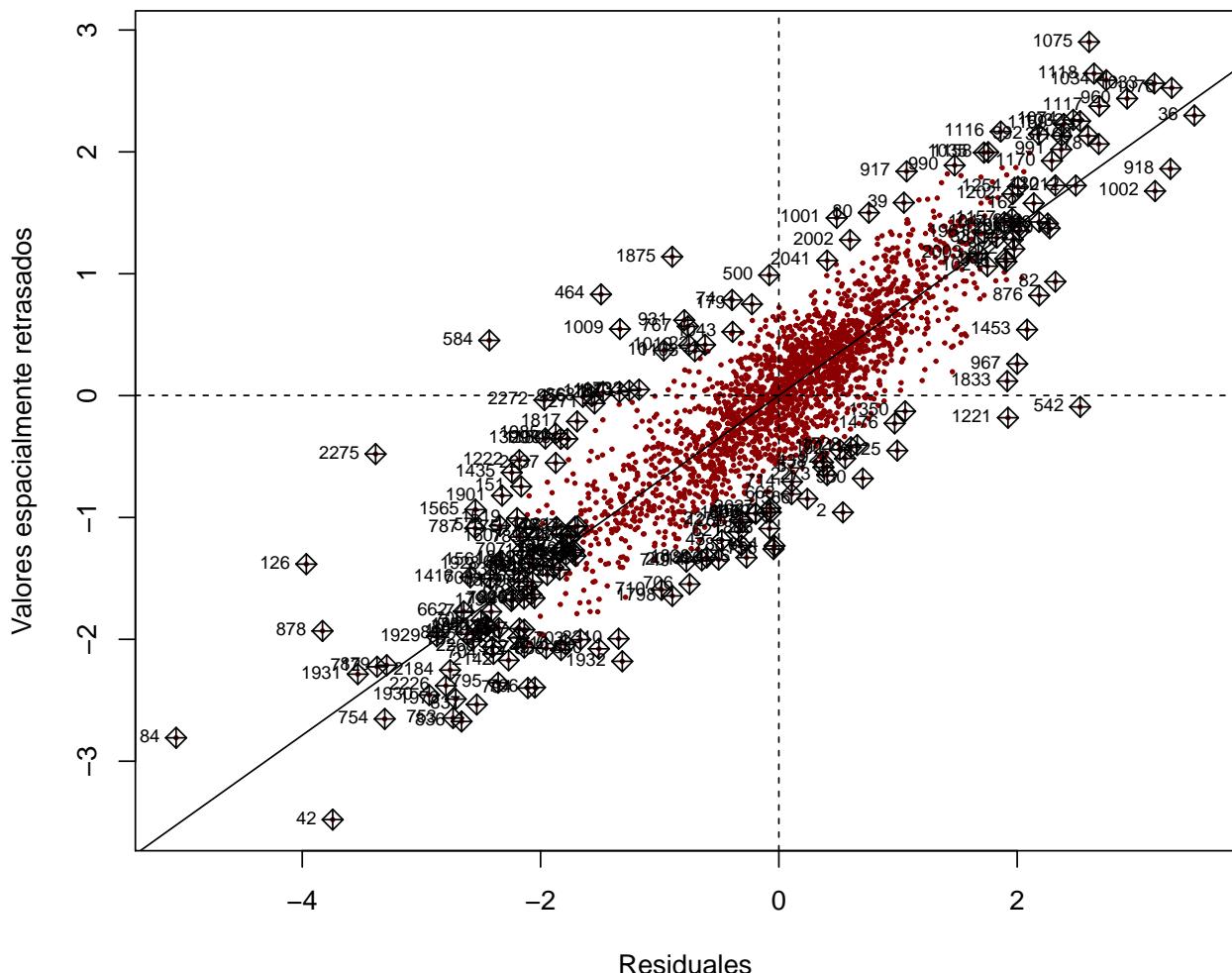
```
moranPlotC1 <- moran.plot(x = datosRindeMaiz$resAjusteC1.M1[-NAs],  
                           listw = nb2listw(nbC1),  
                           labels = as.character(1:nrow(datosRindeMaiz[-NAs])),  
                           pch = 19,  
                           cex = 0.3,  
                           col = "darkred",  
                           ylab = "Valores espacialmente retrasados",  
                           xlab = "Residuales",  
                           main = "C1")
```

C1



```
moranPlotC2 <- moran.plot(x = datosRindeMaiz$resAjusteC2.M1,
  listw = nb2listw(nbC2),
  labels = as.character(1:nrow(datosRindeMaiz)),
  pch = 19,
  cex = 0.3,
  col = "darkred",
  ylab = "Valores espacialmente retrasados",
  xlab = "Residuales",
  main = "C2")
```

C2



```
# Filas de puntos influyentes
ptosInfC1 <- as.numeric(rownames(summary(moranPlotC1)))
```

```

## Potentially influential observations of
## lm(formula = wx ~ x) :
##
##          dfb.1_ dfb.x dffit    cov.r   cook.d hat
## 2      -0.06 -0.09 -0.11_*  0.99_*  0.01    0.00
## 14     0.06 -0.06  0.09_*  0.99_*  0.00    0.00
## 26     0.01 -0.02   0.02   1.00_*  0.00    0.00
## 30     0.05 -0.11  0.12_*  1.00    0.01    0.00
## 31    -0.04   0.01 -0.05   1.00_*  0.00    0.00
## 39     0.06 -0.15  0.16_*  1.00_*  0.01    0.00_*
## 40     0.01 -0.03   0.04   1.00_*  0.00    0.00_*
## 41     0.01 -0.04   0.04   1.01_*  0.00    0.01_*
## 42    -0.08   0.26 -0.27_*  0.99_*  0.04    0.00_*
## 81    -0.07 -0.12 -0.14_*  0.99_*  0.01    0.00
## 83    -0.05   0.05 -0.07   1.00_*  0.00    0.00
## 84     0.10 -0.41  0.42_*  0.99_*  0.09    0.01_*
## 132    0.09 -0.21  0.23 *  0.99_*  0.03    0.00_*

```

## 168	-0.01	-0.02	-0.02	1.00_*	0.00	0.00
## 248	0.10	-0.19	0.21_*	0.98_*	0.02	0.00
## 303	0.01	0.01	0.02	1.00_*	0.00	0.00
## 347	0.01	0.02	0.03	1.00_*	0.00	0.00
## 391	0.00	0.00	0.01	1.00_*	0.00	0.00
## 430	0.06	0.03	0.06	1.00_*	0.00	0.00
## 462	0.00	-0.01	-0.01	1.00_*	0.00	0.00
## 472	-0.05	-0.11	-0.12_*	1.00	0.01	0.00
## 477	0.05	-0.05	0.08	1.00_*	0.00	0.00
## 503	0.00	0.01	0.01	1.00_*	0.00	0.00
## 505	0.05	-0.08	0.09_*	1.00_*	0.00	0.00
## 527	-0.07	0.03	-0.07	0.99_*	0.00	0.00
## 528	0.01	-0.03	0.03	1.00_*	0.00	0.00_*
## 532	0.00	0.00	0.00	1.00_*	0.00	0.00
## 568	0.01	-0.01	0.01	1.00_*	0.00	0.00
## 570	0.00	0.01	-0.01	1.00_*	0.00	0.00_*
## 571	-0.04	0.09	-0.10_*	1.00	0.00	0.00_*
## 572	-0.06	0.03	-0.07	0.99_*	0.00	0.00
## 573	0.00	0.00	0.00	1.00_*	0.00	0.00
## 574	-0.01	0.03	-0.04	1.00_*	0.00	0.00_*
## 575	0.00	0.00	0.00	1.00_*	0.00	0.00
## 585	0.00	0.01	0.01	1.00_*	0.00	0.00
## 612	-0.06	0.11	-0.13_*	1.00_*	0.01	0.00
## 613	0.06	-0.27	0.28_*	1.01_*	0.04	0.01_*
## 614	-0.05	0.08	-0.09_*	1.00	0.00	0.00
## 616	0.00	0.01	-0.01	1.00_*	0.00	0.00_*
## 618	-0.05	0.03	-0.06	1.00_*	0.00	0.00
## 619	0.00	0.00	0.00	1.00_*	0.00	0.00
## 621	0.01	-0.02	0.02	1.00_*	0.00	0.00_*
## 627	0.00	0.00	0.00	1.00_*	0.00	0.00
## 628	0.01	0.02	0.02	1.00_*	0.00	0.00
## 654	-0.07	0.10	-0.12_*	0.99_*	0.01	0.00
## 655	0.02	-0.07	0.07	1.01_*	0.00	0.01_*
## 656	-0.06	0.09	-0.11_*	1.00_*	0.01	0.00
## 658	-0.05	0.08	-0.09_*	1.00	0.00	0.00
## 659	-0.01	0.01	-0.02	1.00_*	0.00	0.00
## 662	-0.02	0.05	-0.05	1.00_*	0.00	0.00_*
## 663	0.06	-0.21	0.22_*	1.00	0.02	0.01_*
## 664	-0.09	0.00	-0.09_*	0.99_*	0.00	0.00
## 696	0.05	-0.14	0.14_*	1.00	0.01	0.00_*
## 697	-0.07	0.13	-0.15_*	0.99_*	0.01	0.00
## 699	-0.02	0.04	-0.04	1.00_*	0.00	0.00
## 700	-0.01	0.02	-0.02	1.00_*	0.00	0.00
## 705	-0.05	0.08	-0.09_*	1.00	0.00	0.00
## 708	0.00	0.01	-0.01	1.00_*	0.00	0.00
## 709	0.01	-0.03	0.03	1.00_*	0.00	0.00_*
## 710	0.00	-0.01	0.01	1.00_*	0.00	0.00
## 716	-0.05	-0.08	-0.10_*	1.00	0.00	0.00
## 738	-0.06	-0.01	-0.06	0.99_*	0.00	0.00
## 740	-0.01	0.01	-0.01	1.00_*	0.00	0.00_*
## 754	-0.01	0.03	-0.03	1.00_*	0.00	0.00
## 755	0.01	-0.01	0.02	1.00_*	0.00	0.00_*
## 780	-0.13	-0.27	-0.30_*	0.97_*	0.04	0.00
## 781	0.05	-0.09	0.10_*	1.00	0.01	0.00

```

## 783 -0.06 -0.02 -0.06  0.99_*  0.00  0.00
## 786 -0.05 -0.02 -0.05  1.00_*  0.00  0.00
## 789  0.00  0.01 -0.01  1.00_*  0.00  0.00
## 820 -0.06 -0.05 -0.08  0.99_*  0.00  0.00
## 826 -0.06 -0.07 -0.09_* 0.99_*  0.00  0.00
## 830 -0.05 -0.06 -0.08  1.00_*  0.00  0.00
## 869 -0.05 -0.05 -0.07  1.00_*  0.00  0.00
## 944 -0.01 -0.02 -0.02  1.00_*  0.00  0.00
## 950 -0.05 -0.07 -0.09_* 1.00   0.00  0.00
## 958  0.00  0.01  0.01  1.00_*  0.00  0.00
## 959  0.01  0.02  0.02  1.00_*  0.00  0.00_*
## 961 -0.01 -0.02 -0.02  1.00_*  0.00  0.00_*
## 962  0.01  0.01  0.01  1.00_*  0.00  0.00
## 1001 -0.04 -0.10 -0.11_* 1.00   0.01  0.00_*
## 1002  0.10 -0.04  0.11_*  0.98_*  0.01  0.00
## 1003  0.00  0.00  0.00  1.00_*  0.00  0.00
## 1004  0.02  0.05  0.05  1.00_*  0.00  0.00_*
## 1008  0.04 -0.18  0.18_*  1.01_*  0.02  0.01_*
## 1033  0.09  0.02  0.09_*  0.99_*  0.00  0.00
## 1046  0.02  0.05  0.05  1.00_*  0.00  0.00_*
## 1047  0.02  0.05  0.05  1.00_*  0.00  0.00_*
## 1048  0.00  0.00  0.00  1.00_*  0.00  0.00
## 1049  0.08 -0.04  0.09  0.99_*  0.00  0.00
## 1050 -0.09  0.40 -0.41_* 1.00_*  0.08  0.01_*
## 1069  0.00  0.01  0.01  1.00_*  0.00  0.00
## 1073  0.06  0.03  0.07  0.99_*  0.00  0.00
## 1074 -0.08 -0.25 -0.27_* 0.99_*  0.04  0.00_*
## 1075  0.10  0.03  0.10_*  0.98_*  0.01  0.00
## 1076 -0.05 -0.19 -0.20_* 1.00   0.02  0.01_*
## 1077  0.02  0.04  0.04  1.00_*  0.00  0.00
## 1088  0.01  0.03  0.04  1.00_*  0.00  0.00_*
## 1091  0.00  0.01 -0.01  1.02_*  0.00  0.01_*
## 1092 -0.12  0.22 -0.25_* 0.98_*  0.03  0.00
## 1114  0.06 -0.02  0.07  0.99_*  0.00  0.00
## 1130  0.01  0.01  0.01  1.00_*  0.00  0.00
## 1133 -0.04  0.25 -0.25_* 1.01_*  0.03  0.02_*
## 1134  0.02 -0.08  0.08  1.01_*  0.00  0.01_*
## 1135 -0.11  0.09 -0.14_* 0.98_*  0.01  0.00
## 1155  0.06 -0.04  0.07  0.99_*  0.00  0.00
## 1160 -0.05 -0.17 -0.18_* 1.00   0.02  0.00_*
## 1169  0.00  0.01  0.01  1.00_*  0.00  0.00
## 1175 -0.09  0.47 -0.48_* 0.99_*  0.11  0.01_*
## 1176 -0.11  0.49 -0.50_* 0.99_*  0.13  0.01_*
## 1177  0.04 -0.21  0.21_*  1.01_*  0.02  0.01_*
## 1178 -0.07  0.09 -0.11_* 0.99_*  0.01  0.00
## 1191 -0.05 -0.05 -0.07  1.00_*  0.00  0.00
## 1192  0.07 -0.17  0.19_*  0.99_*  0.02  0.00_*
## 1217 -0.04  0.23 -0.24_* 1.01_*  0.03  0.01_*
## 1218  0.02 -0.13  0.13_*  1.02_*  0.01  0.02_*
## 1219 -0.01  0.03 -0.03  1.01_*  0.00  0.01_*
## 1220 -0.05  0.08 -0.09_* 1.00   0.00  0.00
## 1254  0.00  0.00  0.00  1.00_*  0.00  0.00
## 1259 -0.02  0.07 -0.07  1.00_*  0.00  0.00_*
## 1260 -0.11  0.17 -0.20_* 0.98_*  0.02  0.00

```

```

## 1261 -0.01 0.04 -0.04 1.01_* 0.00 0.00_*
## 1282 -0.05 0.03 -0.05 1.00_* 0.00 0.00
## 1284 -0.07 -0.04 -0.08 0.99_* 0.00 0.00
## 1295 0.00 0.00 0.01 1.00_* 0.00 0.00
## 1325 -0.05 0.04 -0.07 1.00_* 0.00 0.00
## 1332 -0.05 -0.05 -0.07 1.00_* 0.00 0.00
## 1366 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1384 0.07 -0.11 0.13_* 0.99_* 0.01 0.00
## 1403 0.05 -0.10 0.12_* 1.00 0.01 0.00
## 1410 -0.07 -0.03 -0.07 0.99_* 0.00 0.00
## 1486 0.02 -0.06 0.06 1.00_* 0.00 0.00_*
## 1527 -0.04 0.01 -0.05 1.00_* 0.00 0.00
## 1528 0.04 -0.11 0.12_* 1.00 0.01 0.00_*
## 1535 0.03 -0.07 0.08 1.00 0.00 0.00_*
## 1536 -0.06 0.02 -0.06 0.99_* 0.00 0.00
## 1571 0.02 -0.04 0.04 1.00_* 0.00 0.00
## 1575 -0.06 -0.02 -0.07 0.99_* 0.00 0.00
## 1578 0.01 -0.02 0.03 1.00_* 0.00 0.00
## 1613 0.01 -0.01 0.01 1.00_* 0.00 0.00_*
## 1619 0.01 -0.04 0.04 1.00_* 0.00 0.00_*
## 1624 0.09 -0.12 0.15_* 0.99_* 0.01 0.00
## 1655 0.03 -0.08 0.09 1.00_* 0.00 0.00_*
## 1661 -0.01 0.03 -0.03 1.00_* 0.00 0.00_*
## 1672 0.01 0.03 0.03 1.00_* 0.00 0.00
## 1673 0.01 0.03 0.03 1.00_* 0.00 0.00
## 1678 0.08 -0.09 0.12_* 0.99_* 0.01 0.00
## 1695 0.05 -0.13 0.14_* 1.00 0.01 0.00_*
## 1696 0.01 -0.03 0.03 1.00_* 0.00 0.00_*
## 1697 -0.02 0.04 -0.05 1.00_* 0.00 0.00_*
## 1701 -0.01 0.01 -0.01 1.00_* 0.00 0.00
## 1702 -0.05 0.04 -0.07 1.00_* 0.00 0.00
## 1703 0.03 -0.09 0.09_* 1.00 0.00 0.00_*
## 1704 -0.06 0.05 -0.08 1.00_* 0.00 0.00
## 1706 0.04 -0.11 0.12_* 1.00 0.01 0.00_*
## 1714 0.01 0.02 0.03 1.00_* 0.00 0.00
## 1728 0.05 -0.08 0.10_* 1.00_* 0.00 0.00
## 1737 -0.06 -0.02 -0.06 1.00_* 0.00 0.00
## 1738 -0.05 0.05 -0.07 1.00_* 0.00 0.00
## 1739 0.03 -0.07 0.08 1.00 0.00 0.00_*
## 1743 0.03 -0.07 0.07 1.00 0.00 0.00_*
## 1750 0.00 -0.01 0.01 1.00_* 0.00 0.00_*
## 1751 0.03 -0.09 0.09_* 1.00_* 0.00 0.00_*
## 1759 0.00 -0.01 -0.01 1.00_* 0.00 0.00_*
## 1770 0.06 -0.11 0.12_* 1.00_* 0.01 0.00
## 1780 0.07 -0.15 0.17_* 0.99_* 0.01 0.00
## 1781 -0.08 0.03 -0.08 0.99_* 0.00 0.00
## 1785 0.01 -0.02 0.02 1.00_* 0.00 0.00
## 1790 -0.10 -0.15 -0.19_* 0.98_* 0.02 0.00
## 1801 0.01 0.03 0.03 1.00_* 0.00 0.00_*
## 1803 0.01 0.01 0.01 1.00_* 0.00 0.00
## 1816 0.09 -0.10 0.13_* 0.99_* 0.01 0.00
## 1822 -0.06 -0.02 -0.06 0.99_* 0.00 0.00
## 1843 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1844 0.01 0.01 0.01 1.00_* 0.00 0.00_*

```

```

## 1845 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1857 0.07 -0.07 0.10_* 0.99_* 0.01 0.00
## 1858 -0.08 -0.09 -0.12_* 0.99_* 0.01 0.00
## 1900 0.05 -0.06 0.08 1.00_* 0.00 0.00
## 1930 -0.06 0.02 -0.06 0.99_* 0.00 0.00
## 1950 0.04 -0.08 0.09_* 1.00 0.00 0.00
## 1988 0.05 -0.06 0.08 1.00_* 0.00 0.00
## 2026 -0.05 0.00 -0.05 1.00_* 0.00 0.00
## 2027 0.08 -0.19 0.21_* 0.99_* 0.02 0.00_*
## 2037 0.07 -0.11 0.13_* 0.99_* 0.01 0.00
## 2038 -0.06 -0.05 -0.08 1.00_* 0.00 0.00
## 2080 0.05 -0.07 0.09_* 1.00_* 0.00 0.00
## 2099 0.05 -0.10 0.11_* 1.00 0.01 0.00
## 2121 -0.05 -0.03 -0.06 1.00_* 0.00 0.00
## 2125 0.05 -0.05 0.07 1.00_* 0.00 0.00
## 2141 0.04 -0.08 0.09_* 1.00 0.00 0.00

ptosInfC2 <- as.numeric(rownames(summary(moranPlotC2)))

## Potentially influential observations of
## lm(formula = wx ~ x) :
##
##      dfb.1_ dfb.x dffit cov.r cook.d hat
## 1     0.06 -0.09 0.11_* 1.00_* 0.01 0.00
## 2    -0.08 -0.04 -0.09 0.99_* 0.00 0.00
## 10    0.06 -0.06 0.08 0.99_* 0.00 0.00
## 22    0.05 -0.03 0.06 1.00_* 0.00 0.00
## 36   -0.01 -0.03 -0.03 1.01_* 0.00 0.01_*
## 37    0.03  0.07 0.07 1.00 0.00 0.00_*
## 39    0.05  0.05 0.07 1.00_* 0.00 0.00
## 42   -0.05  0.20 -0.20_* 1.00_* 0.02 0.01_*
## 43   -0.05  0.02 -0.05 1.00_* 0.00 0.00
## 74    0.06 -0.02 0.06 0.99_* 0.00 0.00
## 78    0.01  0.03 0.03 1.00_* 0.00 0.00_*
## 80    0.05  0.04 0.07 1.00_* 0.00 0.00
## 82   -0.04 -0.09 -0.10_* 1.00 0.01 0.00_*
## 84    0.04 -0.22 0.22_* 1.01_* 0.02 0.01_*
## 92   -0.05 -0.02 -0.05 1.00_* 0.00 0.00
## 102   -0.01 -0.02 -0.02 1.00_* 0.00 0.00
## 120   0.01  0.01 0.02 1.00_* 0.00 0.00_*
## 125   -0.06 -0.07 -0.09_* 0.99_* 0.00 0.00
## 126   0.08 -0.33 0.34_* 1.00_* 0.06 0.01_*
## 151   0.04 -0.10 0.11_* 1.00 0.01 0.00_*
## 162   0.00  0.01 0.01 1.00_* 0.00 0.00_*
## 177   0.01 -0.02 0.03 1.00_* 0.00 0.00
## 204   -0.01 -0.02 -0.03 1.00_* 0.00 0.00
## 221   0.00  0.00 0.00 1.00_* 0.00 0.00
## 222   0.01 -0.01 0.02 1.00_* 0.00 0.00
## 337   0.00  0.00 0.00 1.00_* 0.00 0.00
## 438   -0.05 -0.02 -0.05 1.00_* 0.00 0.00
## 464   0.11 -0.17 0.20_* 0.98_* 0.02 0.00
## 500   0.06 -0.01 0.06 0.99_* 0.00 0.00
## 524   0.00  0.01 -0.01 1.00_* 0.00 0.00
## 530   -0.07 -0.05 -0.08 0.99_* 0.00 0.00
## 542   -0.11 -0.28 -0.30_* 0.98_* 0.04 0.00_*

```

## 566	0.00	0.01	-0.01	1.00_*	0.00	0.00
## 571	-0.05	-0.02	-0.05	1.00_*	0.00	0.00
## 573	0.03	-0.08	0.08	1.00	0.00	0.00_*
## 584	0.12	-0.31	0.34_*	0.97_*	0.06	0.00_*
## 621	-0.05	0.02	-0.05	1.00_*	0.00	0.00
## 657	-0.05	0.01	-0.05	1.00_*	0.00	0.00
## 662	0.00	-0.01	0.01	1.00_*	0.00	0.00_*
## 663	0.00	-0.01	0.01	1.00_*	0.00	0.00
## 664	-0.07	0.00	-0.07	0.99_*	0.00	0.00
## 668	-0.05	-0.01	-0.05	1.00_*	0.00	0.00
## 702	-0.01	0.02	-0.02	1.00_*	0.00	0.00_*
## 703	-0.05	0.08	-0.10_*	1.00	0.00	0.00
## 704	-0.03	0.06	-0.07	1.00_*	0.00	0.00_*
## 705	-0.01	0.02	-0.02	1.00_*	0.00	0.00_*
## 706	-0.06	0.05	-0.07	0.99_*	0.00	0.00
## 707	0.01	-0.03	0.03	1.00_*	0.00	0.00_*
## 708	0.01	-0.03	0.03	1.00_*	0.00	0.00_*
## 710	-0.05	0.05	-0.07	1.00_*	0.00	0.00
## 711	-0.02	0.04	-0.04	1.00_*	0.00	0.00_*
## 712	0.01	-0.03	0.03	1.01_*	0.00	0.01_*
## 713	-0.01	0.01	-0.01	1.00_*	0.00	0.00
## 714	-0.04	-0.01	-0.04	1.00_*	0.00	0.00
## 732	0.00	0.00	0.00	1.00_*	0.00	0.00
## 734	-0.01	0.02	-0.02	1.00_*	0.00	0.00
## 744	-0.01	0.01	-0.01	1.00_*	0.00	0.00_*
## 745	-0.04	0.08	-0.09_*	1.00	0.00	0.00
## 747	-0.02	0.06	-0.06	1.00	0.00	0.00_*
## 749	-0.05	0.04	-0.06	1.00_*	0.00	0.00
## 751	-0.02	0.05	-0.06	1.00_*	0.00	0.00_*
## 752	-0.01	0.03	-0.03	1.00_*	0.00	0.00_*
## 753	-0.04	0.12	-0.13_*	1.00	0.01	0.00_*
## 754	-0.02	0.07	-0.07	1.01_*	0.00	0.01_*
## 755	-0.01	0.03	-0.03	1.00_*	0.00	0.00_*
## 756	-0.07	0.00	-0.07	0.99_*	0.00	0.00
## 767	0.06	-0.05	0.08	0.99_*	0.00	0.00
## 777	-0.01	0.01	-0.01	1.00_*	0.00	0.00
## 787	0.04	-0.10	0.11_*	1.00	0.01	0.00_*
## 788	-0.06	0.02	-0.06	0.99_*	0.00	0.00
## 789	0.01	-0.03	0.03	1.00_*	0.00	0.00
## 790	0.00	-0.01	0.01	1.00_*	0.00	0.00
## 794	-0.05	0.12	-0.13_*	1.00	0.01	0.00
## 795	-0.04	0.10	-0.11_*	1.00	0.01	0.00_*
## 796	-0.05	0.12	-0.13_*	1.00_*	0.01	0.00
## 818	0.01	-0.01	0.01	1.00_*	0.00	0.00
## 835	0.01	-0.01	0.02	1.00_*	0.00	0.00_*
## 836	-0.05	0.13	-0.14_*	1.00	0.01	0.00_*
## 837	-0.04	0.12	-0.12_*	1.00	0.01	0.00_*
## 838	-0.05	0.09	-0.10_*	1.00	0.00	0.00
## 839	-0.03	0.06	-0.07	1.00	0.00	0.00_*
## 840	-0.06	0.09	-0.11_*	1.00_*	0.01	0.00
## 868	0.06	-0.10	0.12_*	0.99_*	0.01	0.00
## 872	-0.01	-0.02	-0.02	1.00_*	0.00	0.00
## 876	-0.04	-0.09	-0.10_*	1.00	0.00	0.00_*
## 878	0.04	-0.17	0.17_*	1.00_*	0.02	0.01_*

## 879	0.00	-0.02	0.02	1.01_*	0.00	0.01_*
## 881	-0.01	0.02	-0.03	1.00_*	0.00	0.00_*
## 882	-0.01	0.02	-0.02	1.00_*	0.00	0.00_*
## 915	-0.01	-0.03	-0.03	1.00_*	0.00	0.00
## 917	0.06	0.07	0.09_*	0.99_*	0.00	0.00
## 918	-0.02	-0.08	-0.09	1.01_*	0.00	0.01_*
## 922	-0.05	-0.03	-0.06	1.00_*	0.00	0.00
## 924	-0.01	0.02	-0.02	1.00_*	0.00	0.00
## 931	0.07	-0.06	0.09	0.99_*	0.00	0.00
## 956	0.06	-0.11	0.13_*	0.99_*	0.01	0.00
## 960	0.02	0.07	0.07	1.00_*	0.00	0.00_*
## 967	-0.06	-0.14	-0.15_*	1.00_*	0.01	0.00
## 990	0.05	0.08	0.09_*	1.00	0.00	0.00
## 991	0.02	0.05	0.06	1.00_*	0.00	0.00_*
## 992	0.04	0.08	0.09	1.00	0.00	0.00_*
## 993	-0.01	-0.01	-0.01	1.00_*	0.00	0.00_*
## 1001	0.06	0.03	0.07	0.99_*	0.00	0.00
## 1002	-0.03	-0.10	-0.10_*	1.00_*	0.01	0.01_*
## 1009	0.08	-0.12	0.14_*	0.99_*	0.01	0.00
## 1019	0.05	-0.04	0.07	1.00_*	0.00	0.00
## 1032	0.03	0.07	0.08	1.00_*	0.00	0.00_*
## 1033	0.02	0.07	0.07	1.01_*	0.00	0.01_*
## 1034	0.04	0.11	0.12_*	1.00	0.01	0.00_*
## 1035	0.04	0.08	0.09_*	1.00	0.00	0.00
## 1043	0.04	-0.02	0.05	1.00_*	0.00	0.00
## 1044	0.01	0.01	0.01	1.00_*	0.00	0.00
## 1074	0.03	0.08	0.09	1.00	0.00	0.00_*
## 1075	0.06	0.17	0.18_*	1.00_*	0.02	0.00_*
## 1076	0.01	0.04	0.05	1.01_*	0.00	0.01_*
## 1103	0.05	-0.04	0.06	1.00_*	0.00	0.00
## 1116	0.05	0.10	0.11_*	1.00	0.01	0.00
## 1117	0.03	0.08	0.09	1.00_*	0.00	0.00_*
## 1118	0.05	0.13	0.13_*	1.00	0.01	0.00_*
## 1128	0.00	-0.01	-0.01	1.00_*	0.00	0.00
## 1157	0.00	0.01	0.01	1.00_*	0.00	0.00
## 1158	0.04	0.08	0.09_*	1.00	0.00	0.00
## 1159	0.03	0.08	0.09	1.00	0.00	0.00_*
## 1160	0.02	0.05	0.05	1.00_*	0.00	0.00_*
## 1170	0.02	0.04	0.05	1.00_*	0.00	0.00_*
## 1180	0.05	-0.10	0.11_*	1.00	0.01	0.00
## 1187	0.05	-0.08	0.09_*	1.00_*	0.00	0.00
## 1202	0.02	0.03	0.04	1.00_*	0.00	0.00
## 1212	0.00	0.00	0.00	1.00_*	0.00	0.00_*
## 1221	-0.09	-0.18	-0.20_*	0.99_*	0.02	0.00
## 1222	0.06	-0.13	0.14_*	1.00	0.01	0.00_*
## 1250	-0.05	0.02	-0.05	1.00_*	0.00	0.00
## 1254	0.02	0.04	0.04	1.00_*	0.00	0.00
## 1271	0.06	-0.09	0.11_*	1.00_*	0.01	0.00
## 1296	0.00	0.00	0.00	1.00_*	0.00	0.00
## 1309	0.06	-0.12	0.13_*	1.00_*	0.01	0.00
## 1338	-0.01	-0.02	-0.02	1.00_*	0.00	0.00_*
## 1350	-0.05	-0.05	-0.07	1.00_*	0.00	0.00
## 1380	-0.01	-0.03	-0.03	1.00_*	0.00	0.00_*
## 1416	0.02	-0.05	0.05	1.00_*	0.00	0.00_*

```

## 1435 0.05 -0.12 0.14_* 1.00 0.01 0.00_*
## 1449 0.00 0.01 -0.01 1.00_* 0.00 0.00
## 1453 -0.05 -0.11 -0.12_* 1.00 0.01 0.00
## 1459 -0.05 0.01 -0.05 1.00_* 0.00 0.00
## 1476 -0.05 -0.05 -0.07 1.00_* 0.00 0.00
## 1491 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1519 0.03 -0.07 0.07 1.00 0.00 0.00_*
## 1523 0.01 -0.01 0.01 1.00_* 0.00 0.00
## 1561 0.02 -0.04 0.04 1.00_* 0.00 0.00_*
## 1565 0.05 -0.13 0.14_* 1.00 0.01 0.00_*
## 1566 -0.06 -0.01 -0.06 0.99_* 0.00 0.00
## 1607 0.02 -0.05 0.05 1.00_* 0.00 0.00_*
## 1637 0.01 -0.01 0.01 1.00_* 0.00 0.00
## 1687 -0.05 0.00 -0.05 1.00_* 0.00 0.00
## 1711 -0.05 -0.03 -0.06 1.00_* 0.00 0.00
## 1728 -0.05 -0.03 -0.06 1.00_* 0.00 0.00
## 1733 0.05 -0.06 0.08 1.00_* 0.00 0.00
## 1754 0.02 -0.05 0.05 1.00 0.00 0.00_*
## 1791 0.05 -0.01 0.05 1.00_* 0.00 0.00
## 1795 -0.01 0.01 -0.01 1.00_* 0.00 0.00
## 1796 -0.01 0.02 -0.02 1.00_* 0.00 0.00_*
## 1797 -0.01 0.01 -0.02 1.00_* 0.00 0.00
## 1798 -0.06 0.05 -0.08 1.00_* 0.00 0.00
## 1799 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1800 -0.05 0.01 -0.05 1.00_* 0.00 0.00
## 1813 0.05 -0.07 0.09 1.00_* 0.00 0.00
## 1817 0.05 -0.10 0.11_* 1.00_* 0.01 0.00
## 1833 -0.07 -0.14 -0.16_* 0.99_* 0.01 0.00
## 1838 -0.05 0.03 -0.06 1.00_* 0.00 0.00
## 1840 -0.01 0.02 -0.02 1.00_* 0.00 0.00_*
## 1841 -0.01 0.03 -0.03 1.00_* 0.00 0.00_*
## 1842 0.01 -0.01 0.01 1.00_* 0.00 0.00
## 1875 0.10 -0.09 0.14_* 0.98_* 0.01 0.00
## 1884 -0.01 0.01 -0.01 1.00_* 0.00 0.00
## 1886 -0.06 0.00 -0.06 0.99_* 0.00 0.00
## 1889 0.00 0.00 0.00 1.00_* 0.00 0.00_*
## 1890 -0.06 0.03 -0.06 1.00_* 0.00 0.00
## 1901 0.05 -0.11 0.12_* 1.00 0.01 0.00_*
## 1928 0.01 -0.04 0.04 1.00_* 0.00 0.00_*
## 1929 0.00 0.00 0.00 1.01_* 0.00 0.00_*
## 1930 -0.02 0.07 -0.08 1.00_* 0.00 0.00_*
## 1931 0.01 -0.04 0.04 1.01_* 0.00 0.01_*
## 1932 -0.07 0.10 -0.12_* 0.99_* 0.01 0.00
## 1961 0.01 0.01 0.02 1.00_* 0.00 0.00
## 1970 0.00 0.00 0.00 1.00_* 0.00 0.00
## 1972 -0.03 0.07 -0.08 1.00 0.00 0.00_*
## 1973 -0.03 0.10 -0.10_* 1.00 0.01 0.00_*
## 1974 -0.01 0.03 -0.03 1.00_* 0.00 0.00_*
## 1985 0.06 -0.11 0.12_* 1.00_* 0.01 0.00
## 2001 -0.01 -0.01 -0.01 1.00_* 0.00 0.00
## 2002 0.05 0.03 0.06 1.00_* 0.00 0.00
## 2003 0.00 0.00 0.00 1.00_* 0.00 0.00
## 2014 -0.05 0.03 -0.06 1.00_* 0.00 0.00
## 2016 -0.04 0.08 -0.10_* 1.00 0.00 0.00

```

```

## 2027 -0.05 0.00 -0.05 1.00_* 0.00 0.00
## 2041 0.05 0.02 0.05 1.00_* 0.00 0.00
## 2069 0.05 -0.09 0.11_* 1.00 0.01 0.00
## 2142 -0.03 0.08 -0.09 1.00 0.00 0.00_*
## 2184 -0.02 0.05 -0.06 1.00_* 0.00 0.00_*
## 2226 -0.02 0.07 -0.08 1.00_* 0.00 0.00_*
## 2237 0.04 -0.08 0.09_* 1.00 0.00 0.00
## 2267 -0.01 0.03 -0.03 1.00_* 0.00 0.00
## 2268 -0.02 0.06 -0.06 1.00_* 0.00 0.00_*
## 2272 0.08 -0.16 0.17_* 0.99_* 0.02 0.00
## 2273 -0.05 -0.02 -0.06 1.00_* 0.00 0.00
## 2274 -0.04 0.02 -0.05 1.00_* 0.00 0.00
## 2275 0.11 -0.38 0.40_* 0.98_* 0.08 0.01_*
## 2276 -0.06 0.02 -0.07 0.99_* 0.00 0.00
## 2310 -0.06 0.08 -0.10_* 1.00_* 0.01 0.00

# Histograma de distancias de Cook

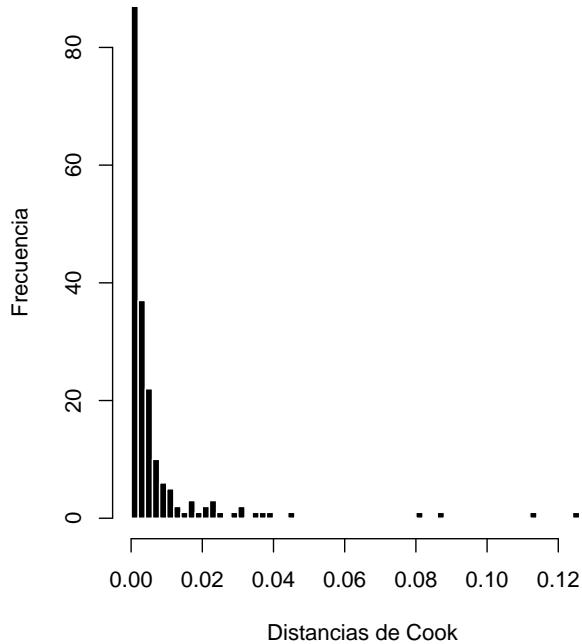
par(mfrow= c(1,2))

hist(moranPlotC1$infmat[ptosInfC1,"cook.d"], seq(0, 0.13, 0.002),
      col = "black", border = "white",
      xlab = "Distancias de Cook", ylab = "Frecuencia",
      main = "C1")

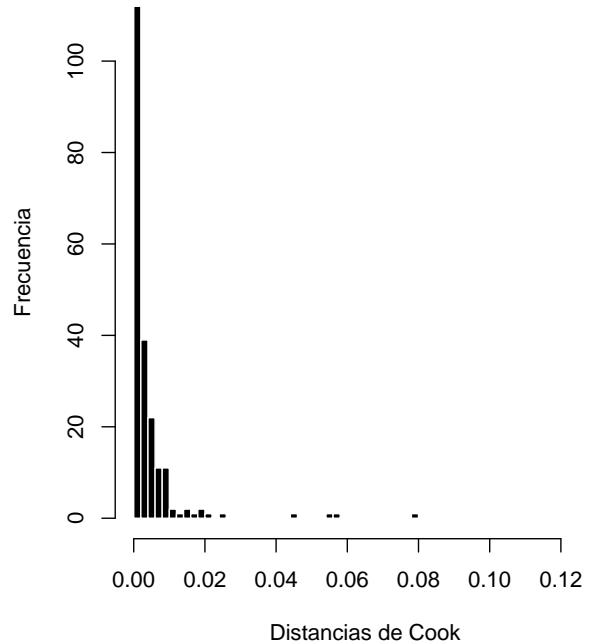
hist(moranPlotC2$infmat[ptosInfC2,"cook.d"], seq(0, 0.13, 0.002),
      col = "black", border = "white",
      xlab = "Distancias de Cook", ylab = "Frecuencia",
      main = "C2")

```

C1



C2



```

par(mfrow=c(1,1))

Se definió considerar como valores atípicos aquellos puntos con una distancia de Cook mayores a  $4/N$ , siendo  $N$  el número total de observaciones.

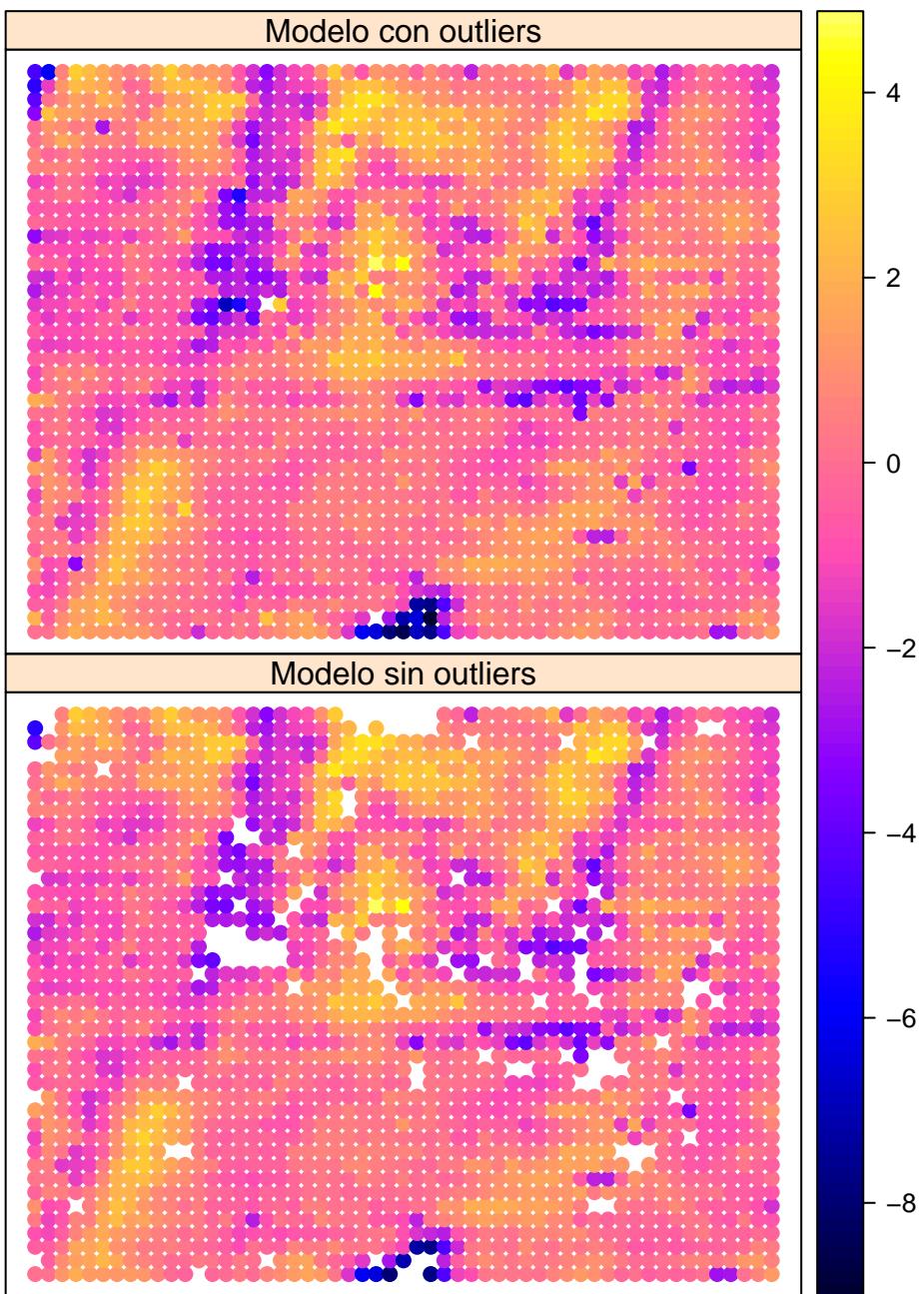
umbralCook <- 4/nrow(datosRindeMaiz)

# Outliers
outliersC1 <- as.numeric(names(which(moranPlotC1$infmat[ptosInfC1, "cook.d"]>umbralCook)))
outliersC2 <- as.numeric(names(which(moranPlotC2$infmat[ptosInfC2, "cook.d"]>umbralCook)))

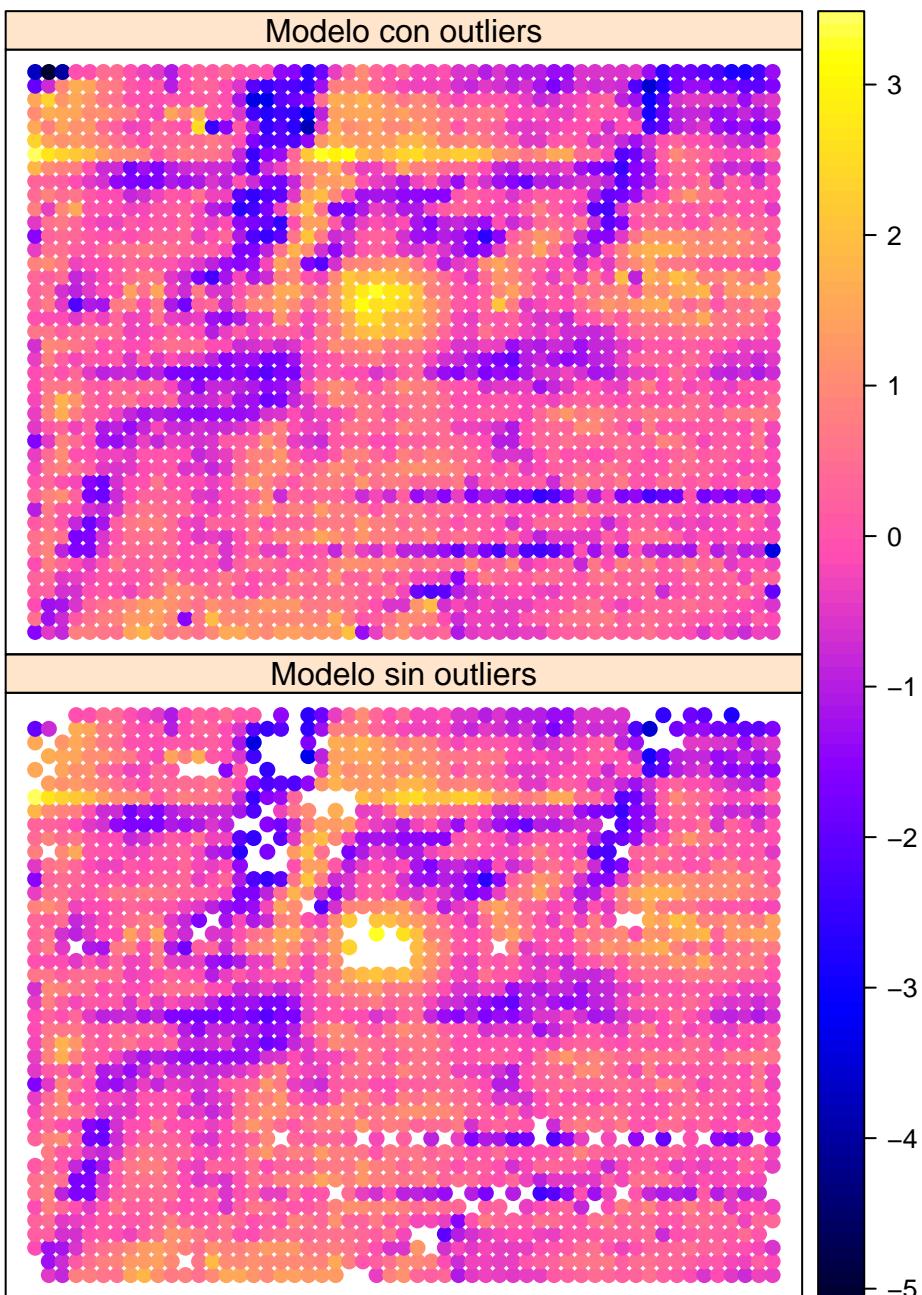
# Agregar valores sin outliers
datosRindeMaiz$ajusteC1.M1.SinOut <- NA
datosRindeMaiz$ajusteC1.M1.SinOut[-outliersC1] <-
  datosRindeMaiz$ajusteC1.M1[-outliersC1]
datosRindeMaiz$resAjusteC1.M1.SinOut <- NA
datosRindeMaiz$resAjusteC1.M1.SinOut[-outliersC1] <-
  datosRindeMaiz$resAjusteC1.M1[-outliersC1]
datosRindeMaiz$ajusteC2.M1.SinOut <- NA
datosRindeMaiz$ajusteC2.M1.SinOut[-outliersC2] <-
  datosRindeMaiz$ajusteC2.M1[-outliersC2]
datosRindeMaiz$resAjusteC2.M1.SinOut <- NA
datosRindeMaiz$resAjusteC2.M1.SinOut[-outliersC2] <-
  datosRindeMaiz$resAjusteC2.M1[-outliersC2]

# Plot C1
spplot(datosRindeMaiz, zcol = c("resAjusteC1.M1.SinOut", "resAjusteC1.M1"),
       names.attr = c("Modelo sin outliers", "Modelo con outliers"), colorkey = TRUE)

```



```
# Plot C2
spplot(datosRindeMaiz, zcol = c("resAjusteC2.M1.SinOut", "resAjusteC2.M1"),
       names.attr = c("Modelo sin outliers", "Modelo con outliers"), colorkey = TRUE)
```

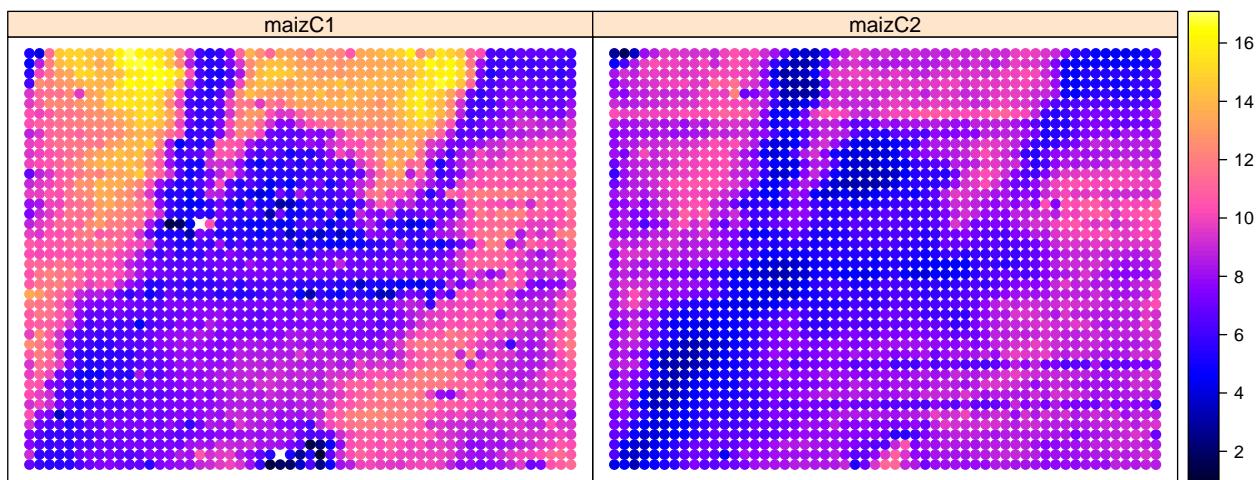


Cambios en la media

Para observar los cambios en la media dentro del lote se realizaron mapas de puntos (“posting”) de los datos originales de *rendimiento de maíz* y las covariables ambientales. No se observaron cambios importantes en la media (valores atípicos) para las covariables ambientales. Sin embargo, para el *rendimiento de maíz* se observaron cambios en la media y algunos fueron eliminados mediante la detección de outliers. Por ejemplo, se observó valores diferentes a los vecinos siguiendo un patrón de línea.

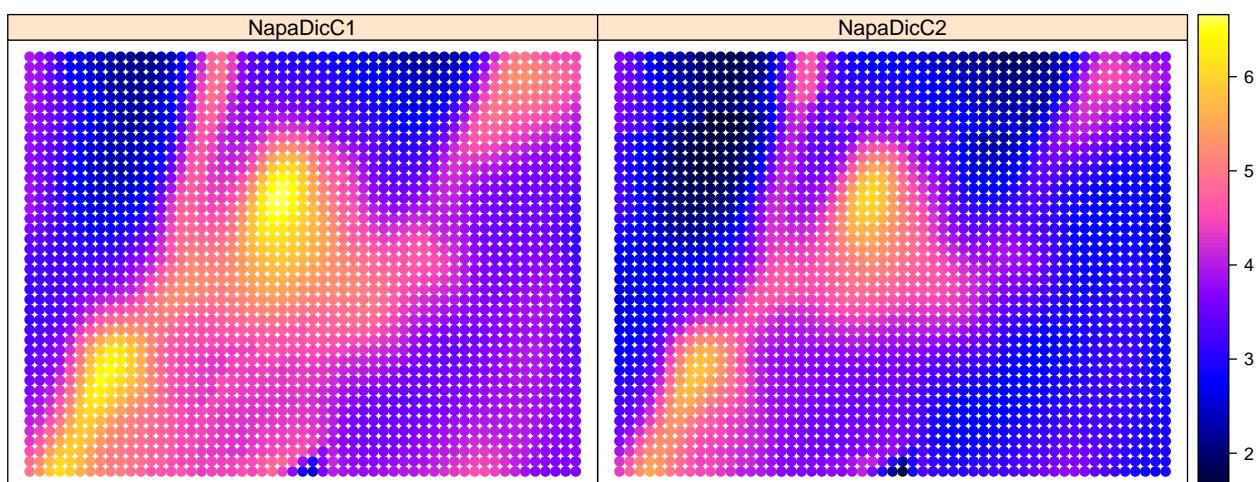
```
# Mapa de puntos de rendimiento de maíz para el lote
spplot(obj = datosRindeMaiz, zcol = c("maizC1", "maizC2"),
       main = "Rendimiento de maíz (t/ha)", colorkey = TRUE)
```

Rendimiento de maíz (t/ha)



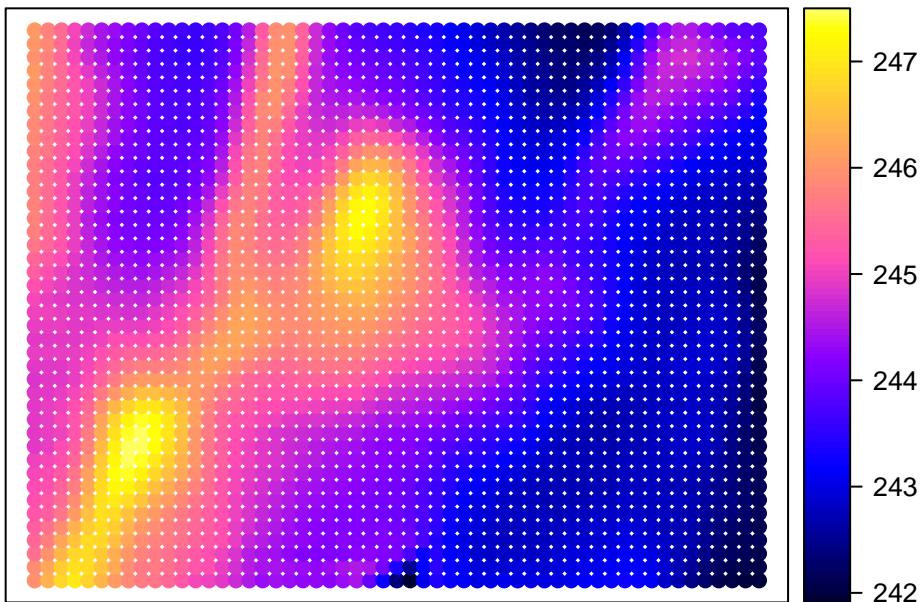
```
# Mapa de puntos de la profundidad de la napa para el lote  
spplot(obj = datosRindeMaiz, zcol = c("NapaDicC1", "NapaDicC2"),  
       main = "Profundidad napa (m)", colorkey = TRUE)
```

Profundidad napa (m)



```
# Mapa de puntos de la elevación del lote  
spplot(obj = datosRindeMaiz, zcol = "Elev",  
       main = "Elevación (m)", colorkey = TRUE)
```

Elevación (m)



Respuesta 2

- 2) Elabore un variograma empírico apropiado para los datos de rinde en ambas campañas por separado. ¿Considera que existe estructura espacial en cada campaña? ¿El proceso espacial observado en cada caso es isotrópico o anisotrópico? ¿La dependencia espacial observada difiere entre campañas?

Variograma empírico

A partir del análisis de los variogramas empíricos puede decirse que existe estructura espacial para las dos campañas ya que:

1. la semivarianza (γ) aumenta con la distancia hasta llegar a una meseta, que implica que la dependencia espacial va disminuyendo con la distancia hasta alcanzar la independencia espacial.
2. al utilizar los residuales de un modelo de tendencia se descarta cualquier efecto de dependencia espacial debido a la misma.

Nota: dado que para estimar los parámetros de ajuste de los modelos teóricos a los variogramas empíricos mediante métodos cuantitativos (*Pregunta 3*) se utilizará la librería **geoR** y para realizar los modelos de predicción-interpolación (*Pregunta 4*), validación cruzada (*Pregunta 5*) y simulaciones (*Pregunta 6*) se utilizará la librería **gstat**, se incluyen los variogramas empíricos y direccionales utilizando las dos librerías (requisitos necesarios de creación de objetos para poder utilizar una y otra librería en las siguientes respuestas).

Con librería **geoR**

```
# Con librería geoR:  
  
dfC1 <- data.frame(as.data.frame(datosRindeMaiz$coords),  
                     "Elev" = datosRindeMaiz$Elev,  
                     "NapaDicC1" = datosRindeMaiz$NapaDicC1,
```

```

"maizC1" = datosRindeMaiz$maizC1)

dfC2 <- data.frame(as.data.frame(datosRindeMaiz@coords),
                     "Elev" = datosRindeMaiz$Elev,
                     "NapaDicC2" = datosRindeMaiz$NapaDicC2,
                     "maizC2" = datosRindeMaiz$maizC2)

# Sacar outliers y NAs
dfC1 <- dfC1[-c(NAs, outliersC1),]
dfC2 <- dfC2[-outliersC2,]

# Objetos geodata
geodataC1 <- as.geodata(obj = dfC1, coords.col = 1:2, data.col = 5,
                           covar.col = c(3,4))
geodataC2 <- as.geodata(obj = dfC2, coords.col = 1:2, data.col = 5,
                           covar.col = c(3,4))

# Crear variograma empírico con el modelo de tendencia elegido
varGeoC1 <- variog(geodataC1, trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                     max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing omnidirectional variogram
varGeoC2 <- variog(geodataC2, trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                     max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

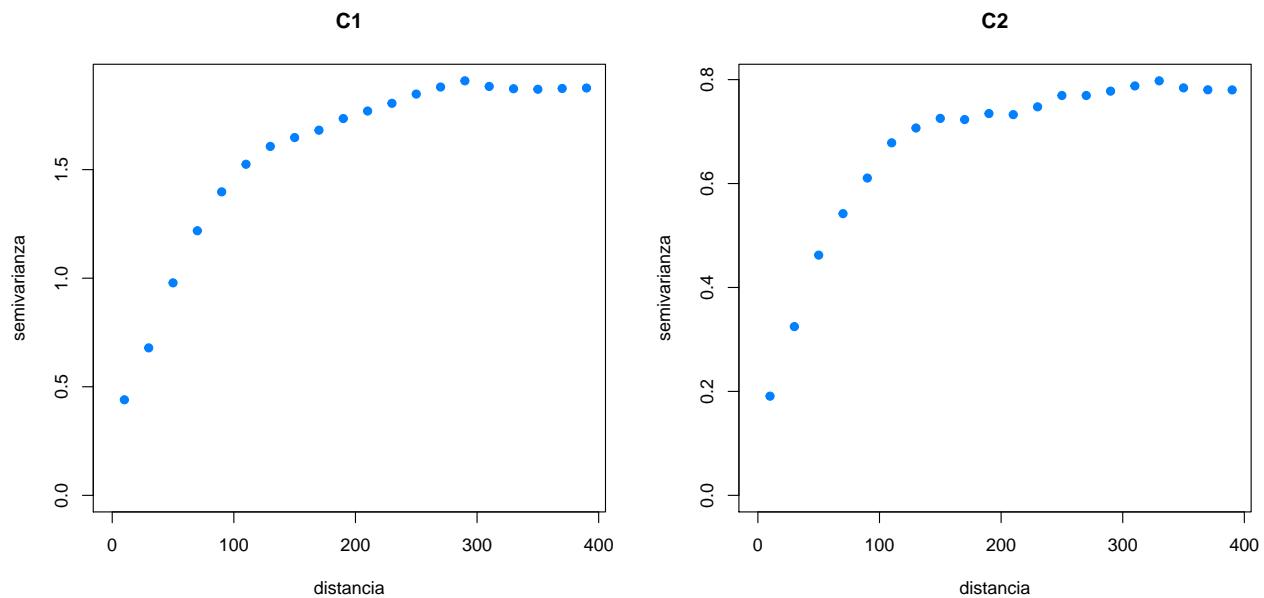
```

variog: computing omnidirectional variogram

```

# Plot variogramas empíricos
par(mfrow = c(1,2))
plot(varGeoC1, pch = 19, col = "#0080FF", ylab = "semivarianza",
      xlab = "distancia", main = "C1")
plot(varGeoC2, pch = 19, col = "#0080FF", ylab = "semivarianza",
      xlab = "distancia", main = "C2")

```

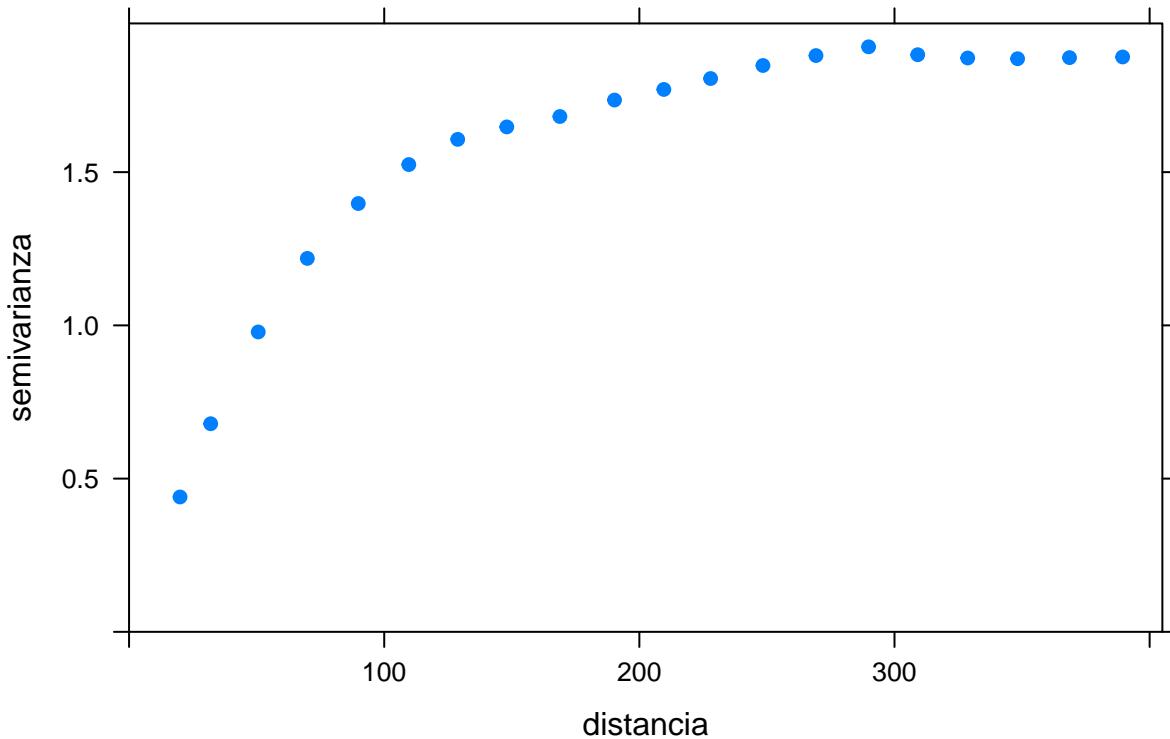


```
par(mfrow = c(1,1))
```

Con librería gstat

```
# Con librería gstat:  
  
# Nota: se utiliza la fórmula "maizC2 ~ NapaDicC2 + Elev" para explicitar el  
# modelo de tendencia utilizado. Podría usarse la variable 'resAjusteC2.M1.SinOut'  
# guardada también en el set de datosRindeMaiz.  
  
# Crear variograma empírico omnidireccional - C1  
varOmniC1 <- variogram(maizC1 ~ NapaDicC1 + Elev,  
                         data = datosRindeMaiz[!is.na(resAjusteC2.M1.SinOut),],  
                         width = 20, cutoff = 400)  
  
# Plotear variograma  
plot(varOmniC1, xlab = "distancia", ylab = "semivarianza",  
     main = "Variograma empírico - C1", pch = 19)
```

Variograma empírico – C1



```
# Con librería gstat:
```

```
# Nota: se utiliza la fórmula "maizC2 ~ NapaDicC2 + Elev" para explicitar el  
# modelo de tendencia utilizado. Podría usarse la variable 'resAjusteC2.M1.SinOut'  
# guardada también en el set de datosRindeMaiz.
```

```
# Crear variograma empírico omnidireccional - C2
```

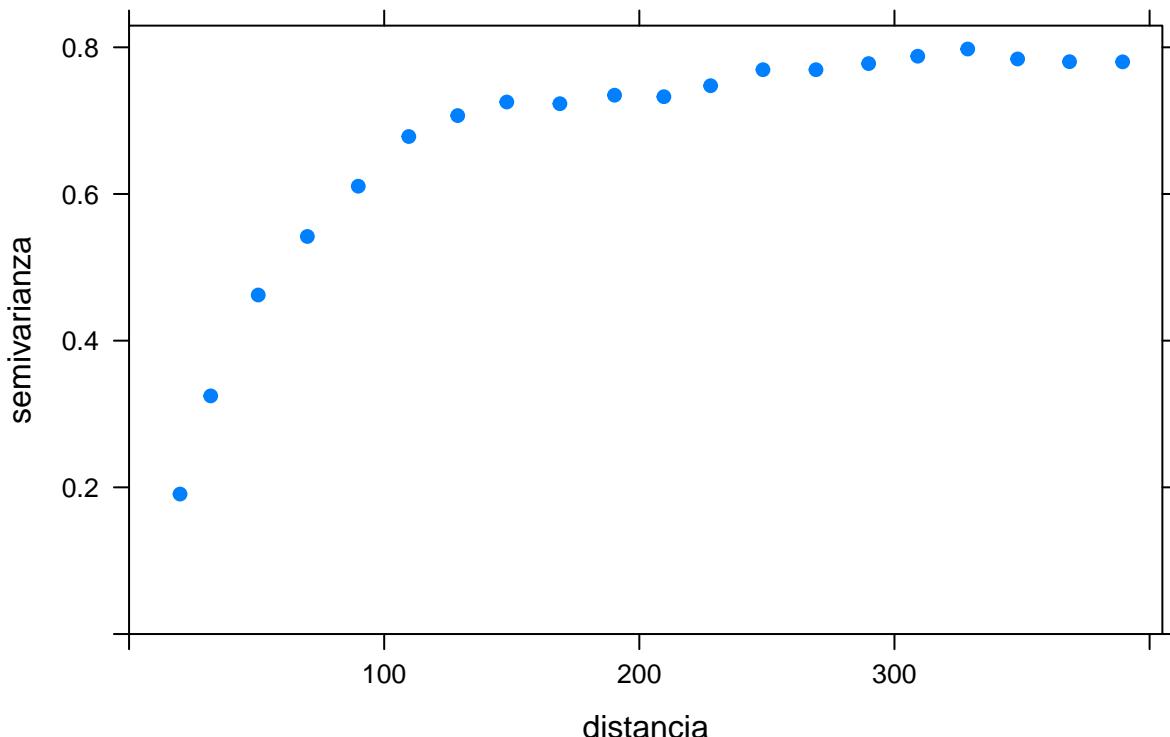
```

varOmniC2 <- variogram(maizC2 ~ NapaDicC2 + Elev,
                        data = datosRindeMaiz[-outliersC2,],
                        width = 20, cutoff = 400)

# Plotear variograma
plot(varOmniC2, xlab = "distancia", ylab = "semivarianza",
      main = "Variograma empírico - C2", pch = 19)

```

Variograma empírico – C2



Proceso espacial: isotropía y anisotropía

A partir de analizar los variogramas direccionales se concluyó que el proceso espacial en las dos campañas es isotrópico. No se observaron diferencias aparentes en los valores de la semivarianza para la cual la misma se estabiliza al alcanzar una meseta ($\gamma_{C1} = 1.7$ $\gamma_{C2} = 0.7$) ni tampoco a la distancia a la cual se llega a γ ($h_{C1} = 100$ y $h_{C2} = 100$) para las distintas direcciones en el espacio ($0^\circ, 45^\circ, 90^\circ$ y 135°).

Con librería geoR

```

# Con librería geoR:

# Crear variogramas direccionales
varGeoC1dir0 <- variog(geodataC1, trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                        direction = 0, tolerance = 45/2, unit.angle = "degrees",
                        max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 0 degrees (0 radians)

```

```

## tolerance angle = 22.5 degrees (0.393 radians)
varGeoC1dir45 <- variog(geodataC1, trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                         direction = 45, tolerance = 45/2, unit.angle = "degrees",
                         max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

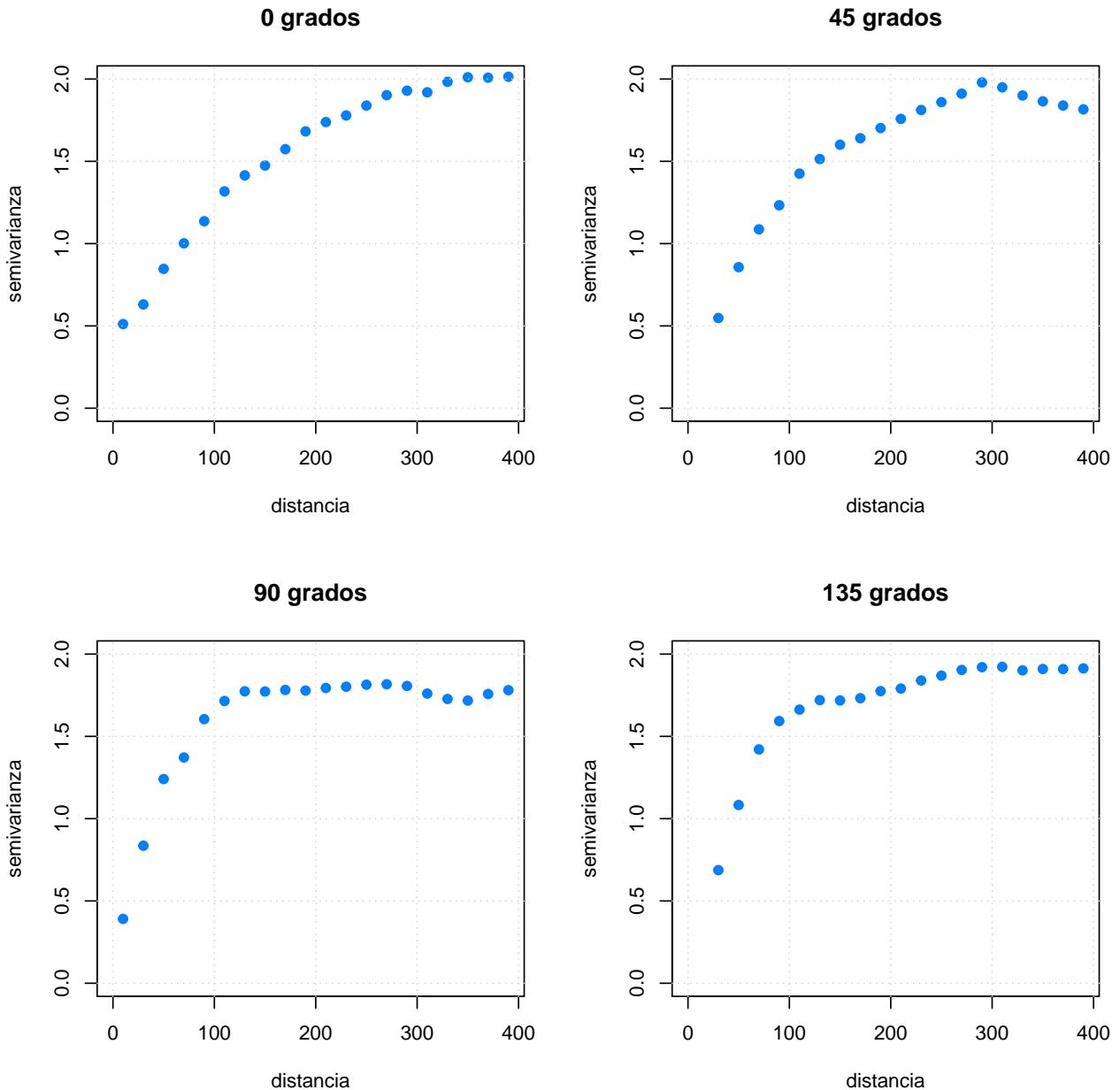
## variog: computing variogram for direction = 45 degrees (0.785 radians)
## tolerance angle = 22.5 degrees (0.393 radians)
varGeoC1dir90 <- variog(geodataC1, trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                         direction = 90, tolerance = 45/2, unit.angle = "degrees",
                         max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 90 degrees (1.571 radians)
## tolerance angle = 22.5 degrees (0.393 radians)
varGeoC1dir135 <- variog(geodataC1, trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                         direction = 135, tolerance = 45/2, unit.angle = "degrees",
                         max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 135 degrees (2.356 radians)
## tolerance angle = 22.5 degrees (0.393 radians)

# Plot variogramas direccionales - C1
par(mfrow = c(2,2))
plot(varGeoC1dir0, pch = 19, col = "#0080FF", ylim = c(0,2),
      ylab = "semivarianza", xlab = "distancia", main = "0 grados")
grid()
plot(varGeoC1dir45, pch = 19, col = "#0080FF", ylim = c(0,2),
      ylab = "semivarianza", xlab = "distancia", main = "45 grados")
grid()
plot(varGeoC1dir90, pch = 19, col = "#0080FF", ylim = c(0,2),
      ylab = "semivarianza", xlab = "distancia", main = "90 grados")
grid()
plot(varGeoC1dir135, pch = 19, col = "#0080FF", ylim = c(0,2),
      ylab = "semivarianza", xlab = "distancia", main = "135 grados")
grid()

```



```

par(mfrow = c(1,1))

# Con librería geoR:

# Crear variogramas direccionales
varGeoC2dir0 <- variog(geodataC2, trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                       direction = 0, tolerance = 45/2, unit.angle = "degrees",
                       max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 0 degrees (0 radians)
##           tolerance angle = 22.5 degrees (0.393 radians)

varGeoC2dir45 <- variog(geodataC2, trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                        direction = 45, tolerance = 45/2, unit.angle = "degrees",
                        max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

```

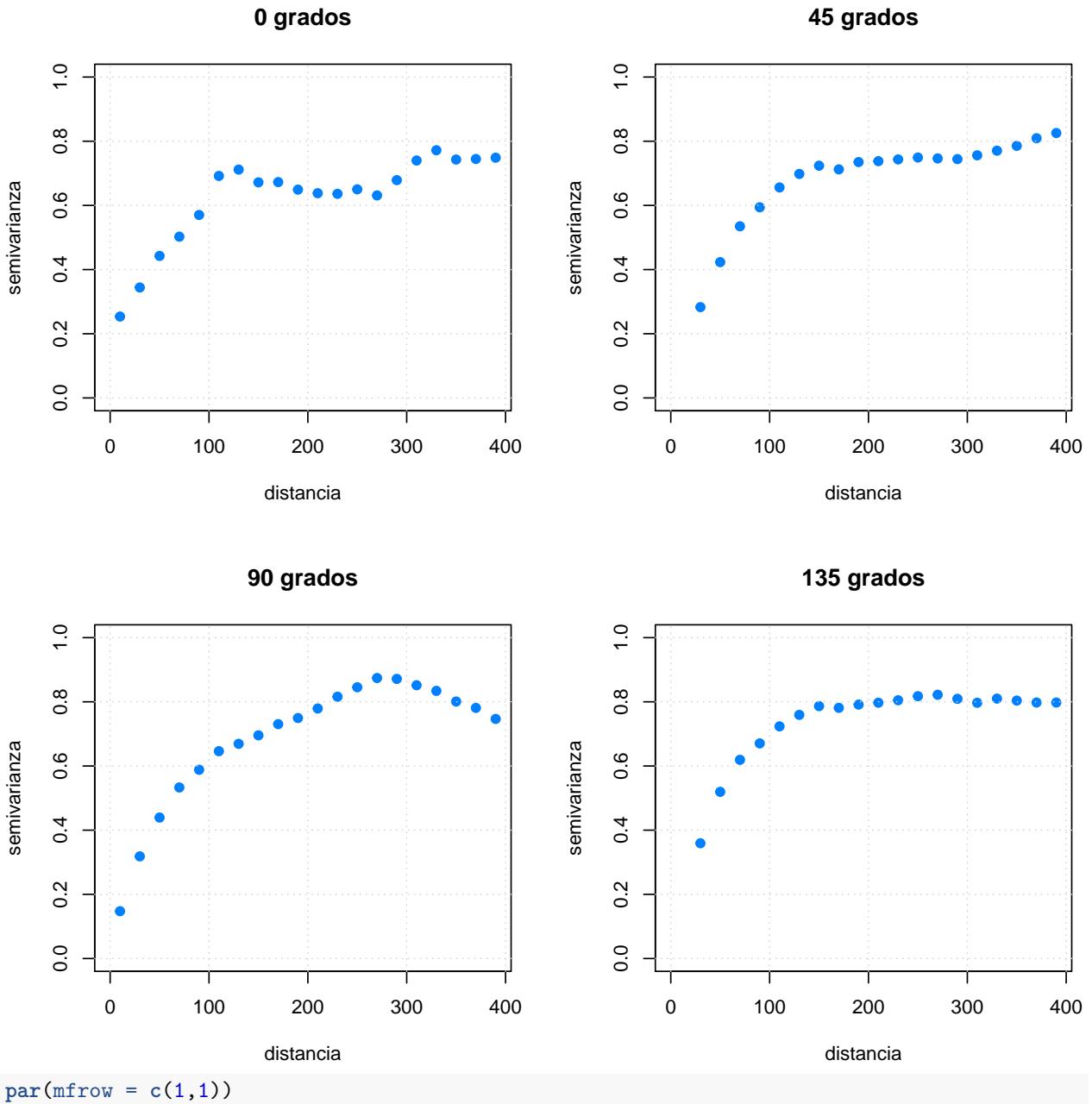
```

## variog: computing variogram for direction = 45 degrees (0.785 radians)
##          tolerance angle = 22.5 degrees (0.393 radians)
varGeoC2dir90 <- variog(geodataC2, trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                        direction = 90, tolerance = 45/2, unit.angle = "degrees",
                        max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 90 degrees (1.571 radians)
##          tolerance angle = 22.5 degrees (0.393 radians)
varGeoC2dir135 <- variog(geodataC2, trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                        direction = 135, tolerance = 45/2, unit.angle = "degrees",
                        max.dist = 400, breaks = seq(from = 0, to = 500, by = 20))

## variog: computing variogram for direction = 135 degrees (2.356 radians)
##          tolerance angle = 22.5 degrees (0.393 radians)
# Plot variogramas direccionales - C2
par(mfrow = c(2,2))
plot(varGeoC2dir0, pch = 19, col = "#0080FF", ylim = c(0,1),
      ylab = "semivarianza", xlab = "distancia", main = "0 grados")
grid()
plot(varGeoC2dir45, pch = 19, col = "#0080FF", ylim = c(0,1),
      ylab = "semivarianza", xlab = "distancia", main = "45 grados")
grid()
plot(varGeoC2dir90, pch = 19, col = "#0080FF", ylim = c(0,1),
      ylab = "semivarianza", xlab = "distancia", main = "90 grados")
grid()
plot(varGeoC2dir135, pch = 19, col = "#0080FF", ylim = c(0,1),
      ylab = "semivarianza", xlab = "distancia", main = "135 grados")
grid()

```

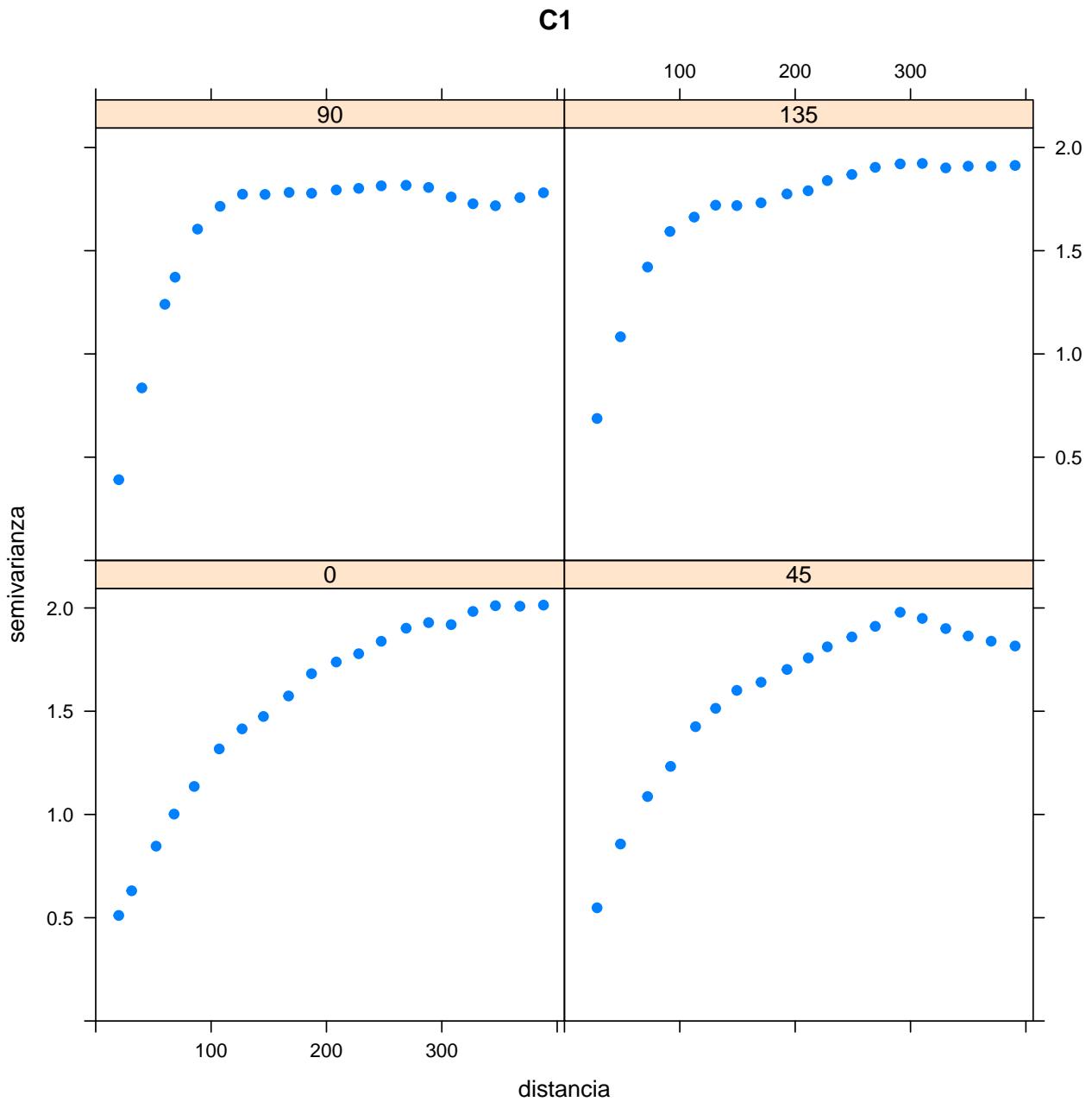


Con librería gstat

```
# Con librería gstat:

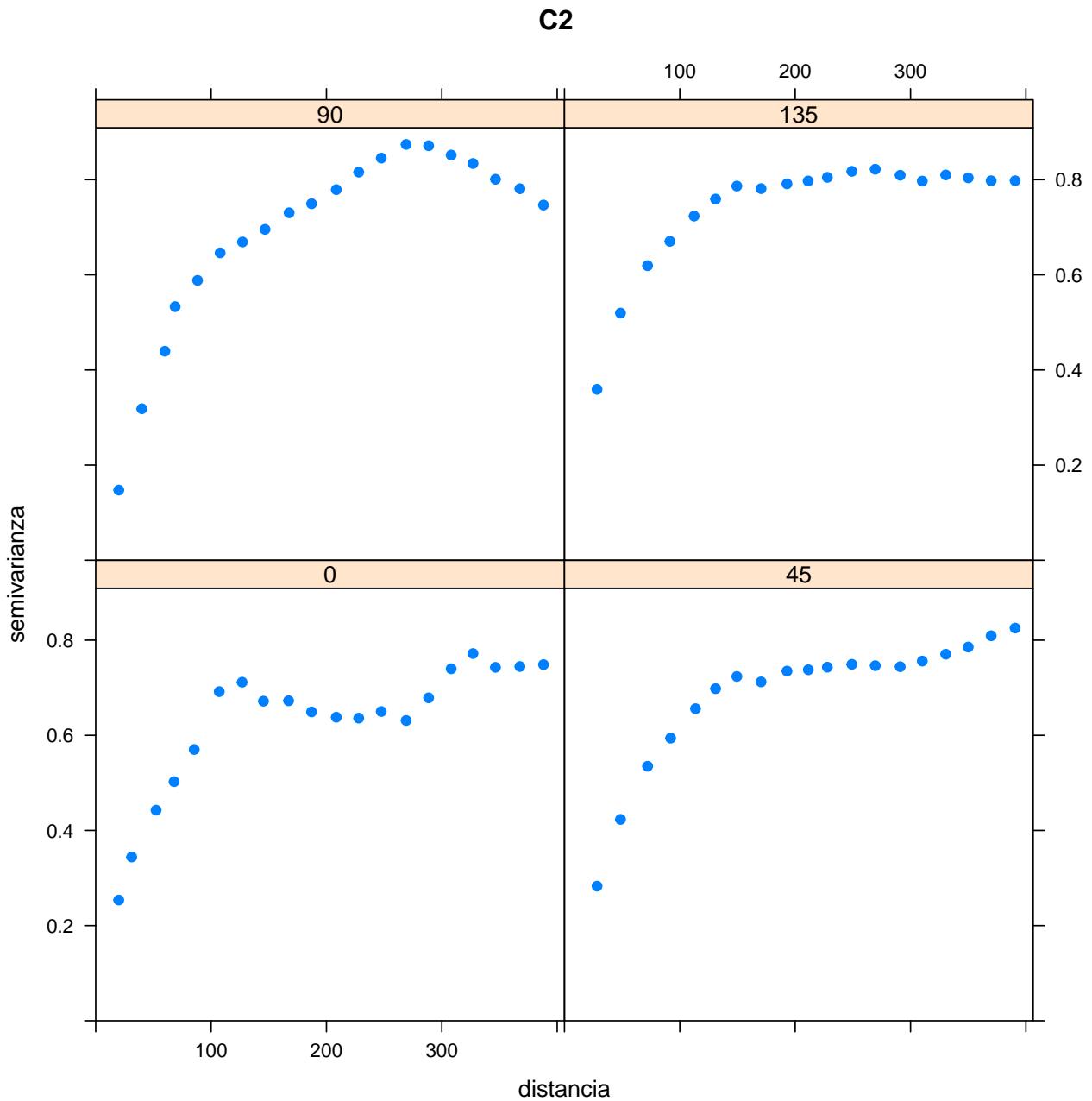
# Crear variograma empírico direccional - C1
varDirC1 <- variogram(maizC1 ~ NapaDicC1 + Elev,
                       data = datosRindeMaiz[-c(NAs,outliersC1),],
                       width = 20, cutoff = 400,
                       alpha = c(0,45,90,135))

# Plotear variograma
plot(varDirC1, xlab = "distancia", ylab = "semivarianza", main = "C1", pch = 19)
```



```
# Con gstat:
varDirC2 <- variogram(maizC2 ~ NapaDicC2 + Elev,
                      data = datosRindeMaiz[-outliersC2,],
                      width = 20, cutoff = 400,
                      alpha = c(0,45,90,135))

# Plotear variograma
plot(varDirC2, xlab = "distancia", ylab = "semivarianza", main = "C2", pch = 19)
```



La dependencia espacial no difiere entre campañas, ya que la distancia (h) a la cual se llega a γ ($h_{C1} = 200$ y $h_{C2} = 200$) es aproximadamente la misma para las dos campañas.

Respuesta 3

- 3) Ajuste al menos tres tipos de modelos permisibles para cada uno de los variogramas muestrales obtenidos en el punto anterior y elija el mejor en cada caso justificando su elección a partir de un criterio cuantitativo objetivo. Utilice procedimientos de estimación estadísticos para ajustar el modelo del variograma.

Ajustar modelos a variograma

Se ajustaron los modelos *esférico*, *exponencial* y *circular* a los variogramas empíricos obtenidos para cada una de las campañas mediante estimación de parámetros por el método de *Máxima verosimilitud restringida (REML)*. Se utilizó para ello la librería **geoR** debido a que presenta la función **likfit** que permite estimar los parámetros necesarios (*rango*, *meseta* y *nugget*) para ajustar un modelo de variograma teórico al variograma empírico y seguir un criterio cuantitativo para seleccionar el mejor modelo. La librería **gstat** posee la función **fit.variogram.reml** que cumple con lo anterior pero solo estima el *rango*. También se probó la función **fit.variogram.gls** pero no es eficaz para muchos datos y requería de un muestreo aleatorio (trabajar con una submuestra) sobre los datos originales.

Para el ajuste de los modelos se asumió que el efecto pepita o nugget fue cero y no se estimó este parámetro.

```
# Se cargan los objetos previamente creados
# para evitar hacer corridas muy demandantes de tiempo o generar otros
# datos debido a muestreos aleatorios
load("~/Documentos/GitHub/IntroductionToGeostatistics/data/RData.RData")

# Tomar muestra para el ajuste de 500 puntos
if(length(which(ls() == "geodataC1")) == 0) {
  geodataC1 <- sample.geodata(geodataC1, size = 500, replace = FALSE)
}
if(length(which(ls() == "geodataC2")) == 0) {
  geodataC2 <- sample.geodata(geodataC2, size = 500, replace = FALSE)
}

# Matriz de posibles parámetros iniciales de rango y meseta para C1
iniCovParsC1 <- cbind("sigma" = seq(1,3,length.out = 20),
                      "phi" = seq(60,240,length.out = 20))

iniCovParsC1 <- as.matrix(iniCovParsC1) # convertir a matriz

if(length(which(ls() == "likfitVarGeoC1.sph")) == 0) {

  # Ajuste modelo esférico
  likfitVarGeoC1.sph <- likfit(geodataC1,
                                 trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                                 ini.cov.pars = iniCovParsC1,
                                 limits = pars.limits(phi = c(50,250), sigmasq = c(0.5,3.5)),
                                 fix.nugget = FALSE,
                                 cov.model = "spherical",
                                 lik.method = "REML")
}

if(length(which(ls() == "likfitVarGeoC1.exp")) == 0) {

  # Ajuste modelo exponencial
  likfitVarGeoC1.exp <- likfit(geodataC1,
                                 trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                                 ini.cov.pars = iniCovParsC1,
                                 limits = pars.limits(phi = c(50,250), sigmasq = c(0.5,3.5)),
                                 fix.nugget = TRUE,
                                 nugget = 0,
                                 cov.model = "exponential",
                                 lik.method = "REML")
```

```

}

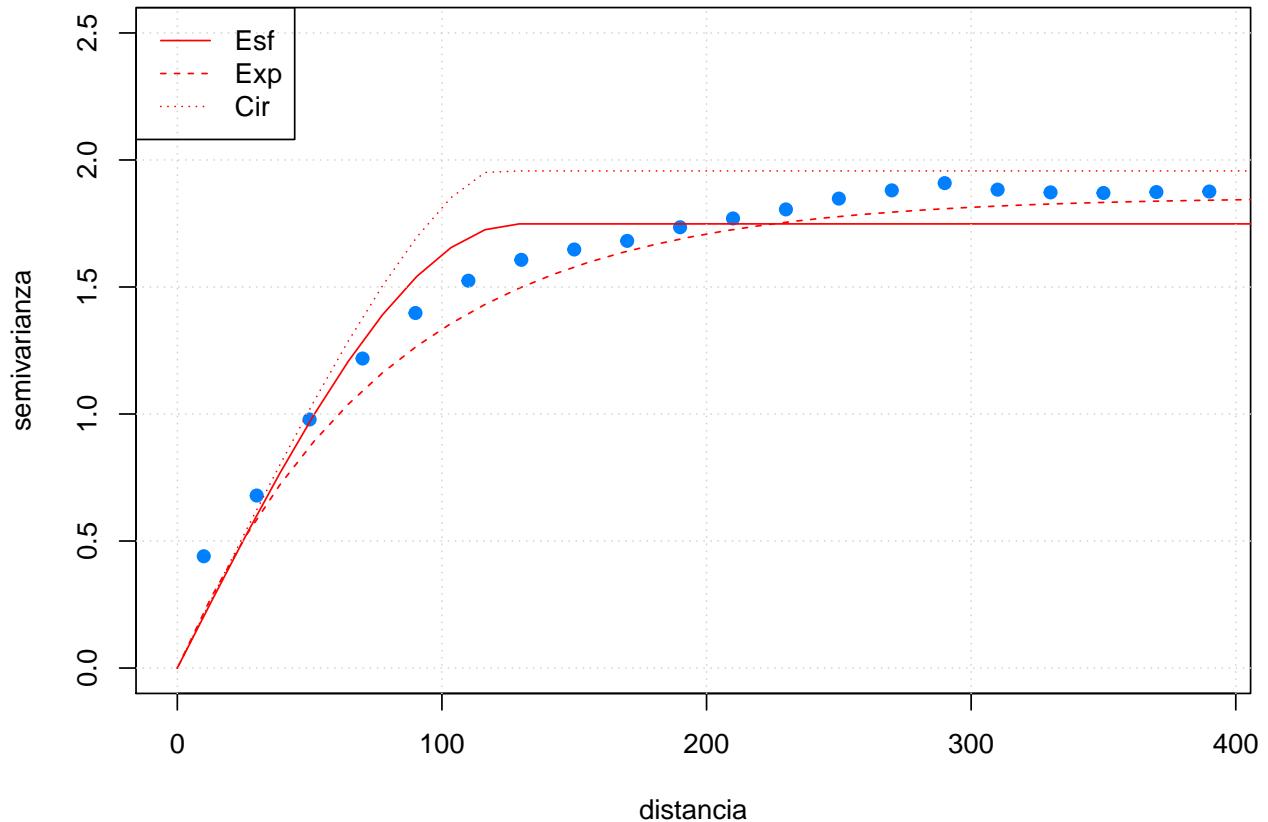
if(length(which(ls() == "likfitVarGeoC1.cir")) == 0) {

  # Ajuste modelo circular
  likfitVarGeoC1.cir <- likfit(geodataC1,
                                trend = trend.spatial(data~NapaDicC1+Elev, geodataC1),
                                ini.cov.pars = iniCovParsC1,
                                limits = pars.limits(phi = c(50,250), sigmasq = c(0.5,3.5)),
                                fix.nugget = TRUE,
                                nugget = 0,
                                cov.model = "circular",
                                lik.method = "REML")
}

# Plot variograma empírico + modelos
plot(varGeoC1, main = "C1", ylab = "semivarianza",
      xlab = "distancia", pch = 19, col = "#0080FF", ylim = c(0,2.5))
lines.variomodel(likfitVarGeoC1.sph, col = "red", lty = 1)
lines.variomodel(likfitVarGeoC1.exp, col = "red", lty = 2)
lines.variomodel(likfitVarGeoC1.cir, col = "red", lty = 3)
grid()
legend('topleft', lty = c(1,2,3), legend = c('Esf','Exp','Cir'), col = rep("red", 3))

```

C1



```

# Selección de modelo por criterio de información de Akaike
AICmodelosC1 <- data.frame("Modelo" = c('Esférico','Exponencial','Circular'),
                            "AIC" = c(likfitVarGeoC1.sph$AIC,
                                     likfitVarGeoC1.exp$AIC,
                                     likfitVarGeoC1.cir$AIC))
print(AICmodelosC1)

##          Modelo      AIC
## 1    Esférico 1322.100
## 2 Exponencial 1325.434
## 3   Circular 1321.998

# Matriz de posibles parámetros iniciales de rango y meseta para C1
iniCovParsC2 <- cbind("sigma" = seq(0.5,1,length.out = 20),
                       "phi" = seq(100,300,length.out = 20))

iniCovParsC2 <- as.matrix(iniCovParsC2) # convertir a matriz

if(length(which(ls() == "likfitVarGeoC2.sph")) == 0) {

  # Ajuste modelo esférico
  likfitVarGeoC2.sph <- likfit(geodataC2,
                                trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                                ini.cov.pars = iniCovParsC2,
                                limits = pars.limits(phi = c(90,310), sigmasq = c(0.4,1.1)),
                                fix.nugget = TRUE,
                                nugget = 0,
                                cov.model = "spherical",
                                lik.method = "REML")
}

if(length(which(ls() == "likfitVarGeoC2.exp")) == 0) {

  # Ajuste modelo exponencial
  likfitVarGeoC2.exp <- likfit(geodataC2,
                                trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                                ini.cov.pars = iniCovParsC2,
                                limits = pars.limits(phi = c(90,310), sigmasq = c(0.4,1.1)),
                                fix.nugget = TRUE,
                                nugget = 0,
                                cov.model = "exponential",
                                lik.method = "REML")
}

if(length(which(ls() == "likfitVarGeoC2.cir")) == 0) {

  # Ajuste modelo circular
  likfitVarGeoC2.cir <- likfit(geodataC2,
                                trend = trend.spatial(data~NapaDicC2+Elev, geodataC2),
                                ini.cov.pars = iniCovParsC2,
                                limits = pars.limits(phi = c(90,310), sigmasq = c(0.4,1.1)),
                                fix.nugget = TRUE,
                                nugget = 0,
}

```

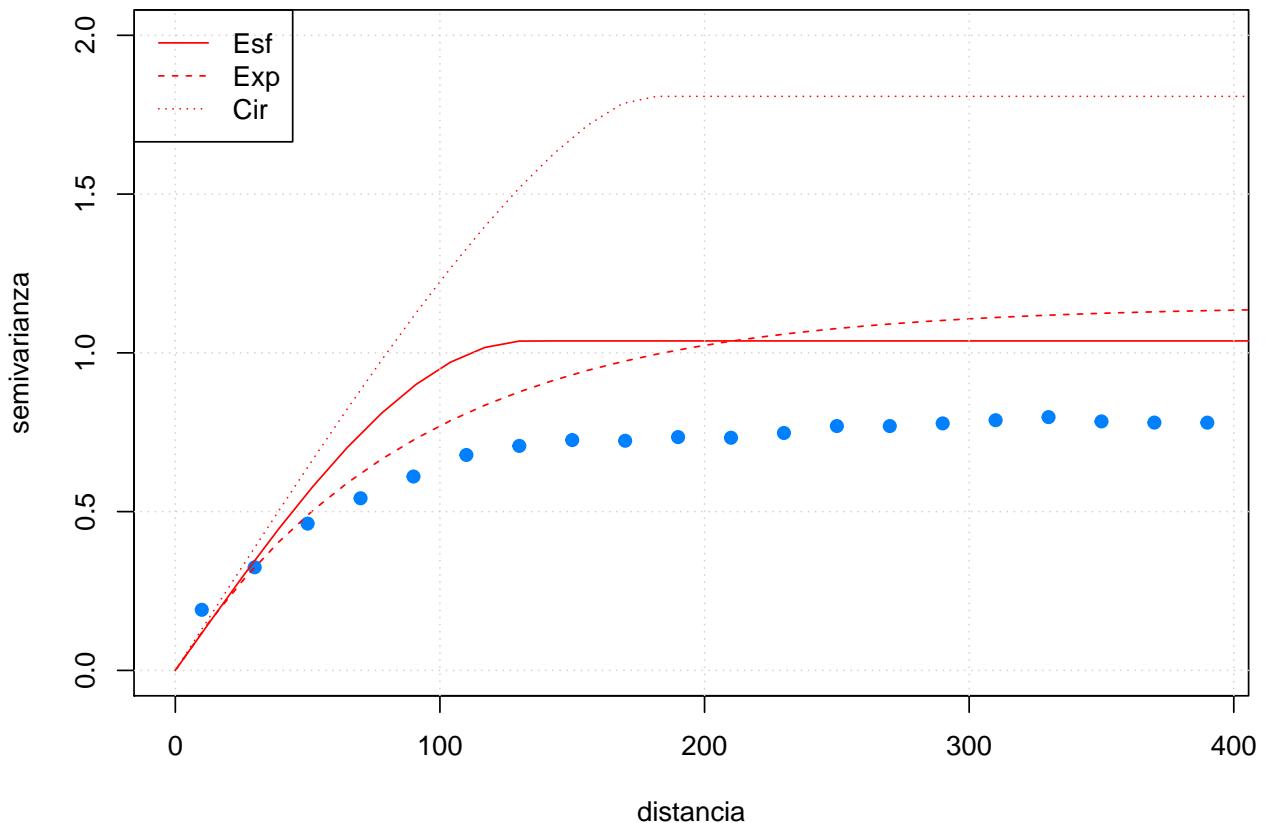
```

        cov.model = "circular",
        lik.method = "REML")
}

# Plot variograma empírico + modelos
plot(varGeoC2, main = "C2", ylab = "semivarianza",
      xlab = "distancia", pch = 19, col = "#0080FF", ylim = c(0,2))
lines.variomodel(likfitVarGeoC2.sph, col = "red", lty = 1)
lines.variomodel(likfitVarGeoC2.exp, col = "red", lty = 2)
lines.variomodel(likfitVarGeoC2.cir, col = "red", lty = 3)
grid()
legend('topleft', lty = c(1,2,3), legend = c('Esf','Exp','Cir'), col = rep("red", 3))

```

C2



```

# Selección de modelo por criterio de información de Akaike
AICmodelosC2 <- data.frame("Modelo" = c('Esférico','Exponencial','Circular'),
                             "AIC" = c(lkifitVarGeoC2.sph$AIC,
                                       lkifitVarGeoC2.exp$AIC,
                                       lkifitVarGeoC2.cir$AIC))
print(AICmodelosC2)

##          Modelo      AIC
## 1    Esférico 1038.402
## 2 Exponencial 1027.014
## 3    Circular 1069.704

```

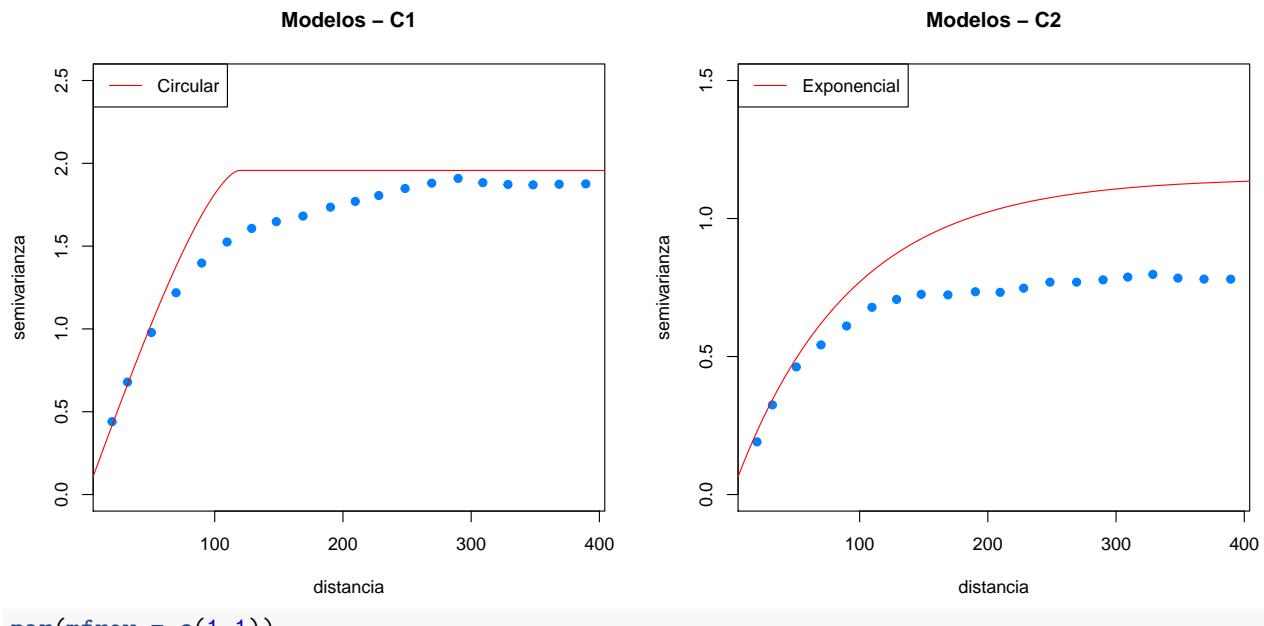
Para la campaña C1 el mejor ajuste del modelo de variograma teórico a los datos fue para el *Circular*

($AICesf_{C1} = 1322.100$, $AICexp_{C1} = 1325.434$ y $AICcir_{C1} = 1321.998$) mientras que para la C2 fue el modelo *Exponencial* ($AICesf_{C2} = 1038.402$, $AICexp_{C2} = 1027.014$ y $AICcir_{C2} = 1069.704$). En ambos casos, se eligió dicho modelo debido a que presentó el *AIC* más bajo.

```
varMejorTeoC1 <- vgm(psill = likfitVarGeoC1.cir$sigmasq,
                      model = "Cir",
                      range = likfitVarGeoC1.cir$phi,
                      nugget = 0)

varMejorTeoC2 <- vgm(psill = likfitVarGeoC2.exp$sigmasq,
                      model = "Exp",
                      range = likfitVarGeoC2.exp$phi,
                      nugget = 0)

# Plotear variograma experimental y ajuste modelo teórico
par(mfrow = c(1,2))
plot(x = varOmniC1$dist, y = varOmniC1$gamma, main = "Modelos - C1",
      col = "#0080FF", ylab = "semivarianza", xlab = "distancia",
      ylim = c(0, 2.5), pch = 19)
lines(variogramLine(object = varMejorTeoC1, maxdist = 500), col = "red", lwd = 1, lty = 1)
legend("topleft", c("Circular"), lty = 1, col = "red")
plot(x = varOmniC2$dist, y = varOmniC2$gamma, main = "Modelos - C2",
      col = "#0080FF", ylab = "semivarianza", xlab = "distancia",
      ylim = c(0, 1.5), pch = 19)
lines(variogramLine(object = varMejorTeoC2, maxdist = 500), col = "red", lwd = 1, lty = 1)
legend("topleft", c("Exponencial"), lty = 1, col = "red")
```



Respuesta 4

- 4) Realice una interpolación para los datos de rinde de cada campaña mediante kriging ordinario puntual y kriging en bloque. Utilice una grilla de predicción hasta la mitad de fina que la originalmente provista para el kriging puntual y un tamaño de bloque que agrupe al menos

nueve observaciones originales. Elabore para cada campaña tanto mapas del rinde de maíz predicho, como de la variabilidad de la predicción. Justifique todas las decisiones necesarias para realizar estas interpolaciones.

Los valores más altos de varianza en todas las predicciones se observaron para aquellos puntos cercanos o sobre celdas donde se eliminaron valores atípicos u outliers.

Tamaño de la grilla y bloque

Para obtener una grilla de predicción de la mitad de fina que la originalmente provista, se utilizó una grilla de con un tamaño de celda de la mitad ($10m$) que los datos originales ($20m$) para realizar el *Kriging Ordinario Puntual*. Además se ajustó la extensión de la grilla de predicción para que ajustara a la original. Para obtener un tamaño de bloque que agrupó al menos nueve observaciones originales (teniendo en cuenta que los puntos originales están separados $20m$ entre si), se utilizó un tamaño de bloque de $60m * 60m$ para realizar el *Kriging en Bloque*.

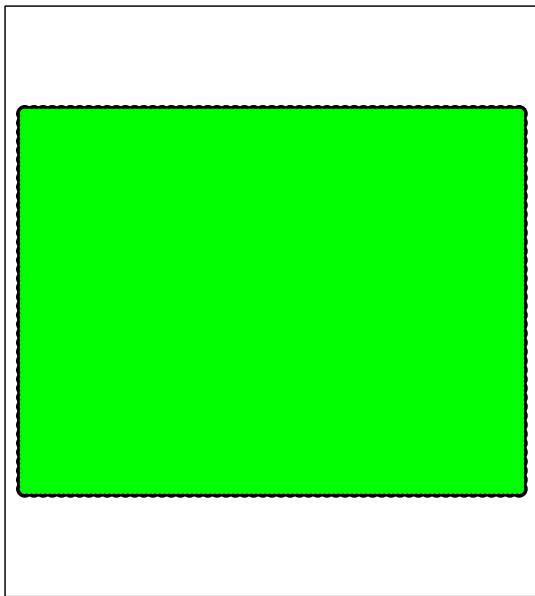
```
# Crear nuevo borde
bb <- matrix(data = c(bbox(datosRindeMaiz)[["Long", "min"]],
                         bbox(datosRindeMaiz)[["Long", "max"]] + 10,
                         bbox(datosRindeMaiz)[["Lat", "min"]],
                         bbox(datosRindeMaiz)[["Lat", "max"]] + 10),
              2, 2, byrow = TRUE)

colnames(bb) <- c("min", "max")
rownames(bb) <- c("Long", "Lat")

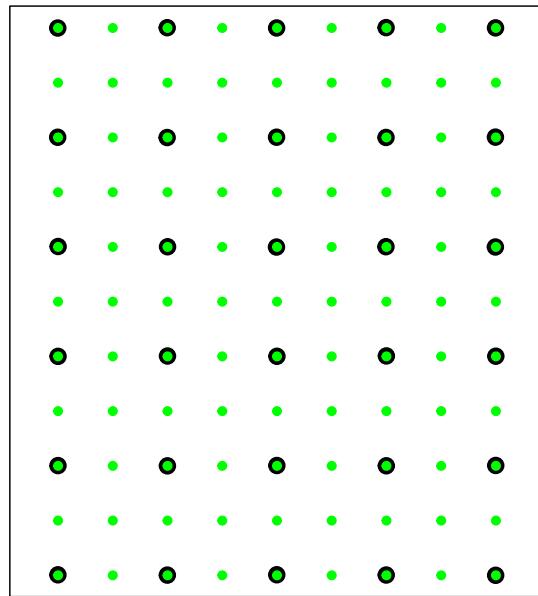
# Definir grilla donde predecir valores
puntosPredecir <- spsample(x = datosRindeMaiz,
                            type = "regular",
                            bb = bb,
                            cellsize = 10,
                            offset = c(0,0))

# Mostrar grilla
par(mfrow = c(1,2))
plot(datosRindeMaiz[,], cex = 0, main = "Todos")
plot(datosRindeMaiz, pch = 19, cex = 1.5, add = TRUE)
plot(puntosPredecir, pch = 19, col = "green", cex = 0.8, add = TRUE)
box()
plot(datosRindeMaiz[505:510,], cex = 0, main = "Detalle")
plot(datosRindeMaiz, pch = 19, cex = 1.5, add = TRUE)
plot(puntosPredecir, pch = 19, col = "green", cex = 0.8, add = TRUE)
box()
```

Todos



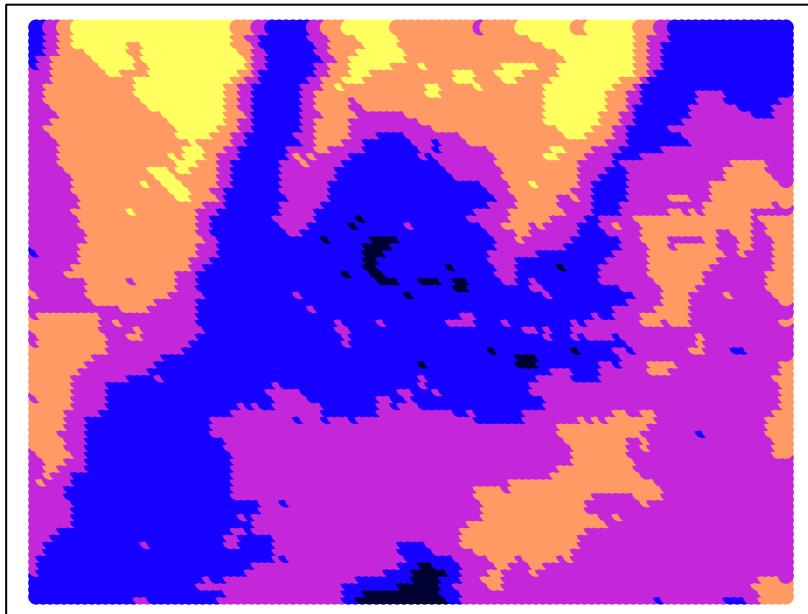
Detalle



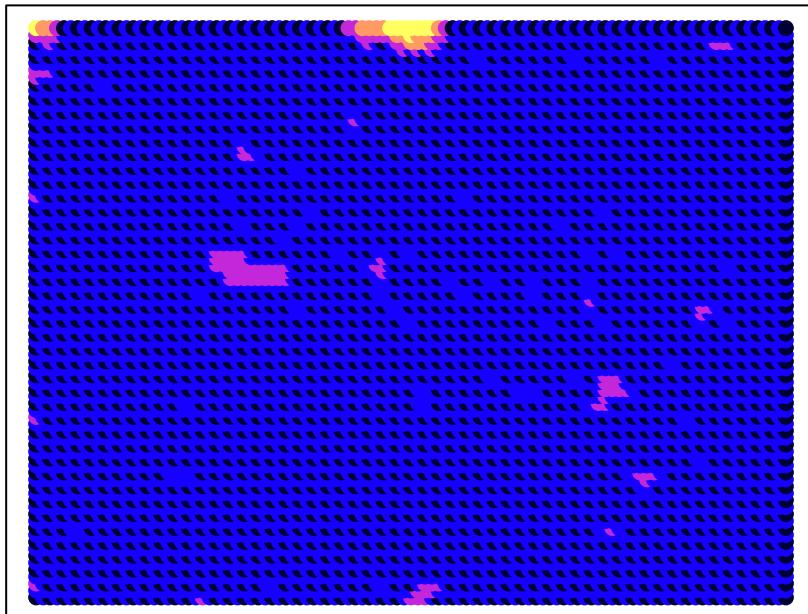
```
par(mfrow = c(1,1))
```

Kriging Ordinario Puntual

```
if(length(which(ls() == "intOK.C1")) == 0) {  
  
  # Interpolacion por Kriging Ordinario (KO)  
  intOK.C1 <- krige(formula = maizC1 ~ 1,  
                     locations = datosRindeMaiz[-c(NAs, outliersC1),],  
                     newdata = puntosPredecir,  
                     model = varMejorTeoC1)  
}  
  
# Predicho  
spplot(intOK.C1, zcol = "var1.pred")
```



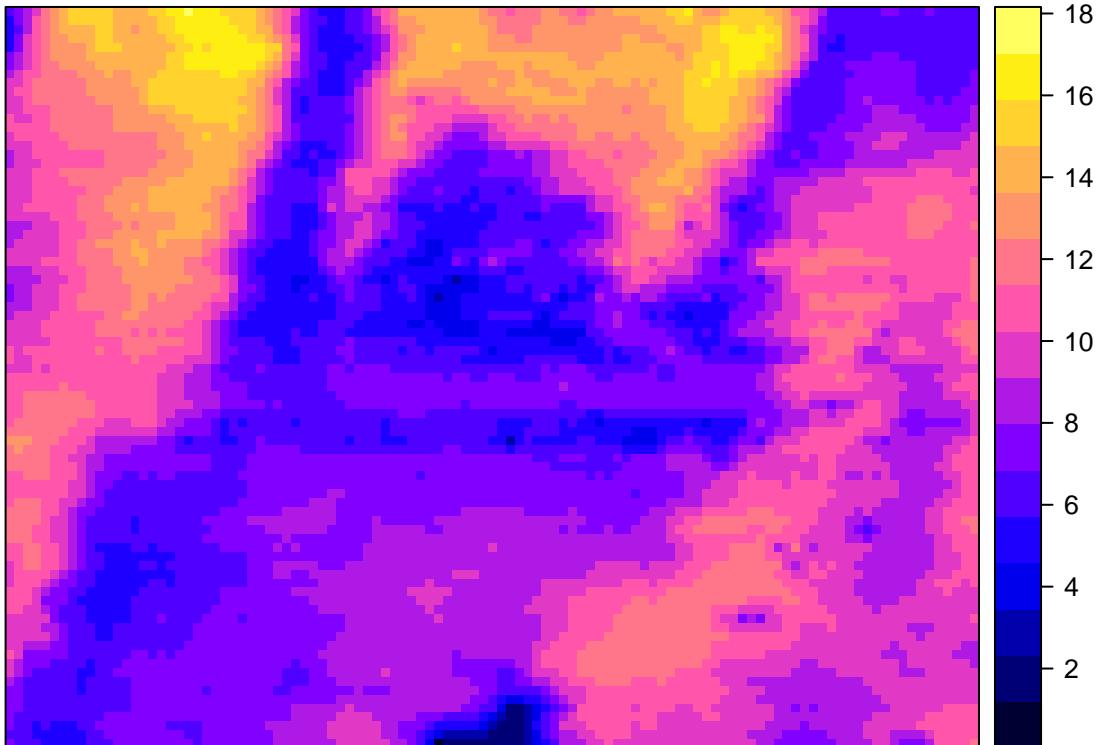
```
spplot(intOK.C1, zcol = "var1.var")
```



```
# Visualizar como grilla
intOK.C1.grid <- SpatialPixelsDataFrame(points = puntosPredecir,
                                         data = intOK.C1@data)

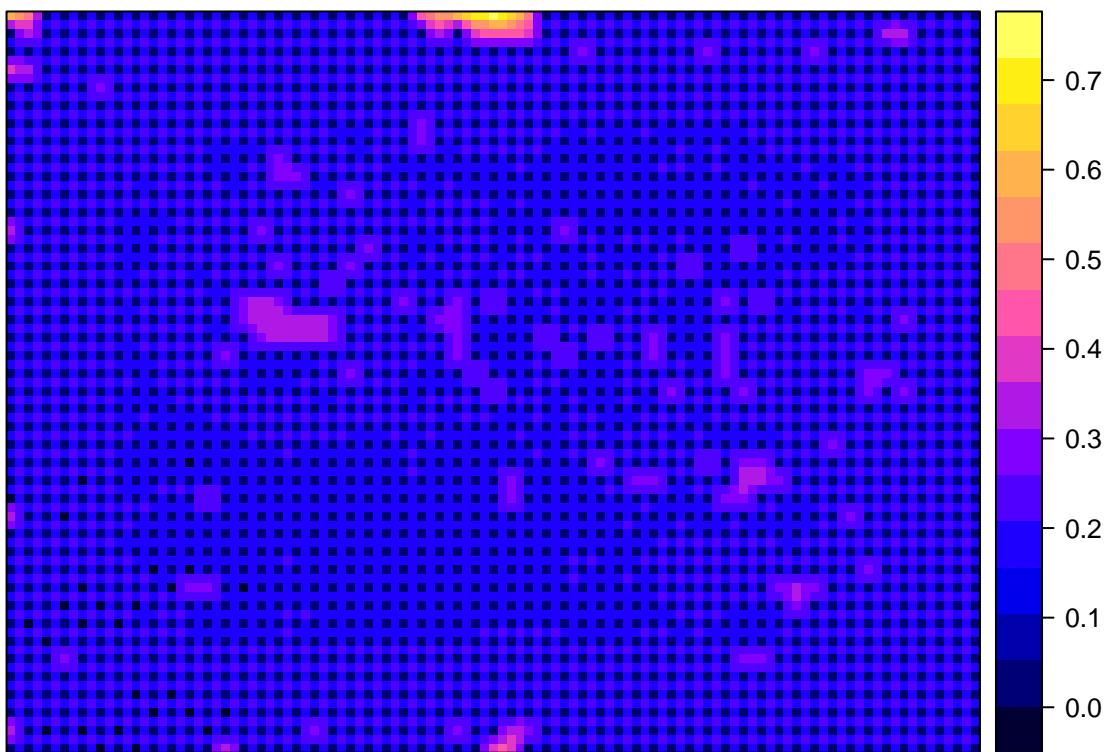
spplot(intOK.C1.grid, main = "Kriging Ordinario C1 - predicción", zcol = "var1.pred")
```

Kriging Ordinario C1 – predicción

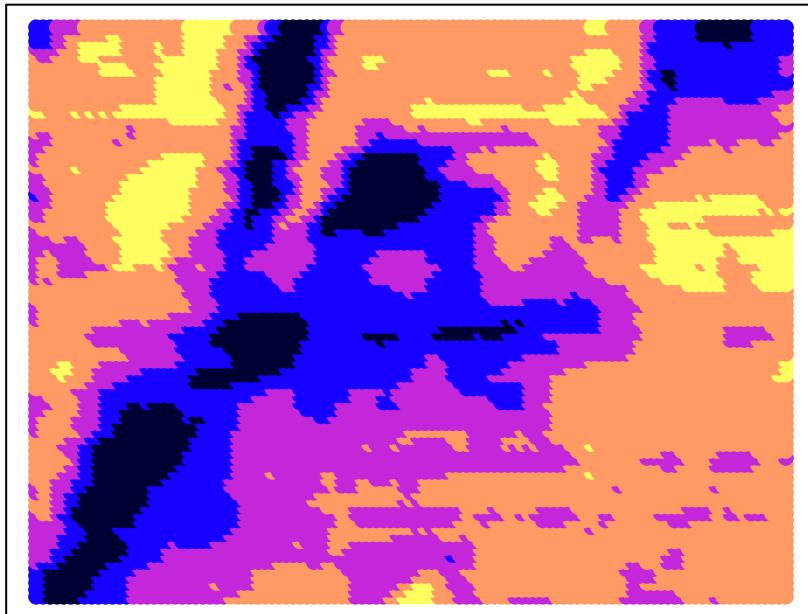


```
spplot(intOK.C1.grid, main = "Kriging Ordinario C1 - varianza", zcol = "var1.var")
```

Kriging Ordinario C1 – varianza

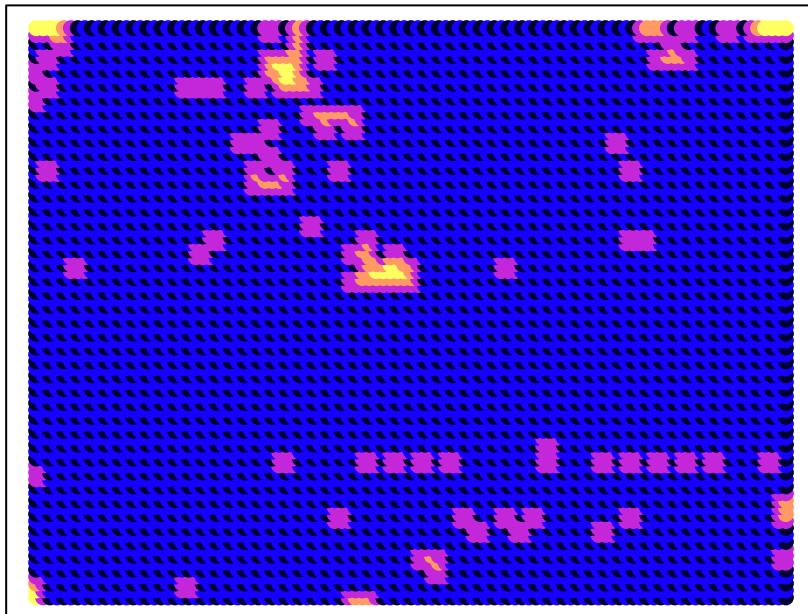


```
if(length(which(ls() == "intOK.C2")) == 0) {  
  
  # Interpolacion por Kriging Ordinario (KO)  
  intOK.C2 <- krig(formula = maizC2 ~ 1,  
                    locations = datosRindeMaiz[-outliersC2,],  
                    newdata = puntosPredecir,  
                    model = varMejorTeoC2)  
}  
  
# Predicho  
spplot(intOK.C2, zcol = "var1.pred")
```



- [2.896, 4.619]
- (4.619, 6.342]
- (6.342, 8.065]
- (8.065, 9.788]
- (9.788, 11.51]

```
spplot(intOK.C2, zcol = "var1.var")
```

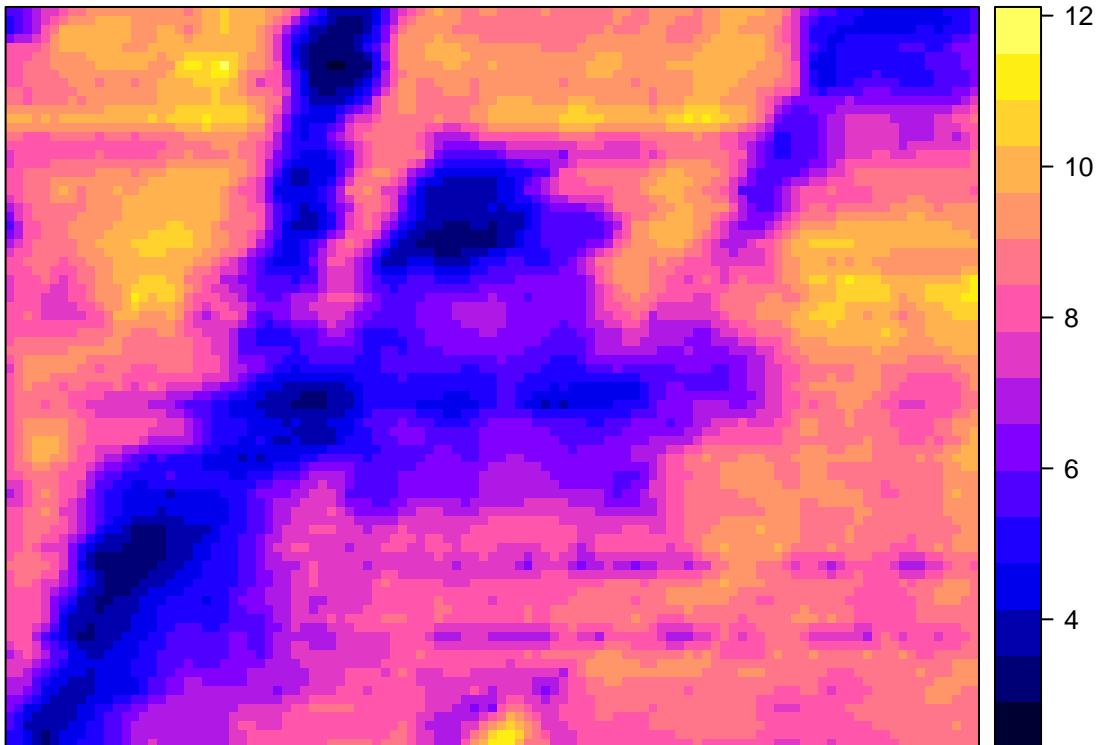


- [5.933e-05, 0.07604]
- (0.07604, 0.152]
- (0.152, 0.228]
- (0.228, 0.304]
- (0.304, 0.38]

```
# Visualizar como grilla
intOK.C2.grid <- SpatialPixelsDataFrame(points = puntosPredecir,
                                         data = intOK.C2@data)

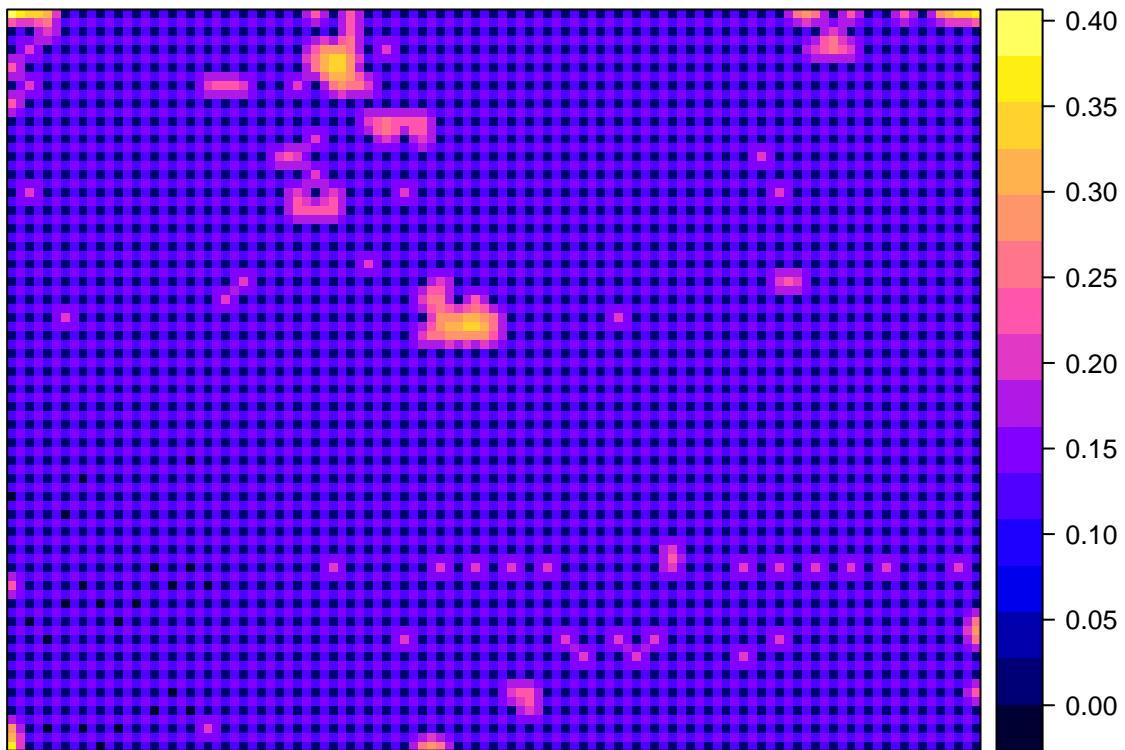
spplot(intOK.C2.grid, main = "Kriging Ordinario C2 - predicción", zcol = "var1.pred")
```

Kriging Ordinario C2 – predicción



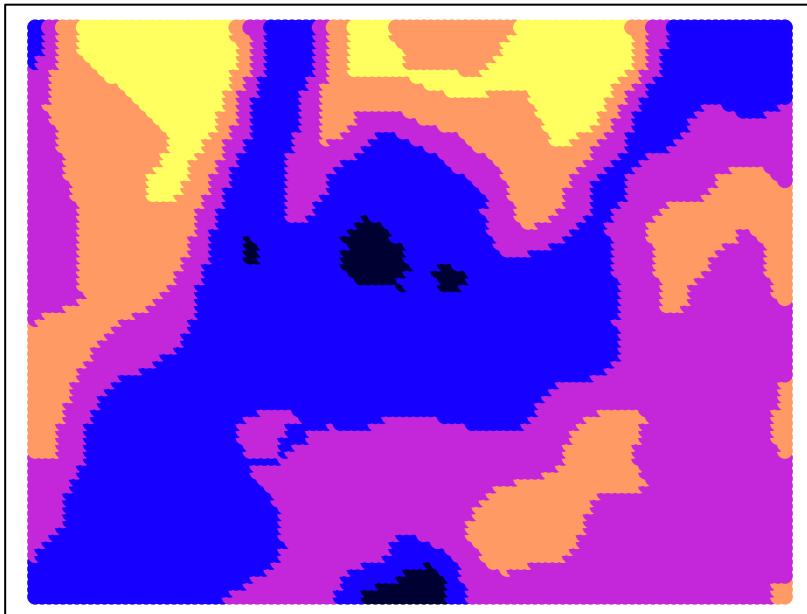
```
spplot(intOK.C2.grid, main = "Kriging Ordinario C2 - varianza", zcol = "var1.var")
```

Kriging Ordinario C2 – varianza



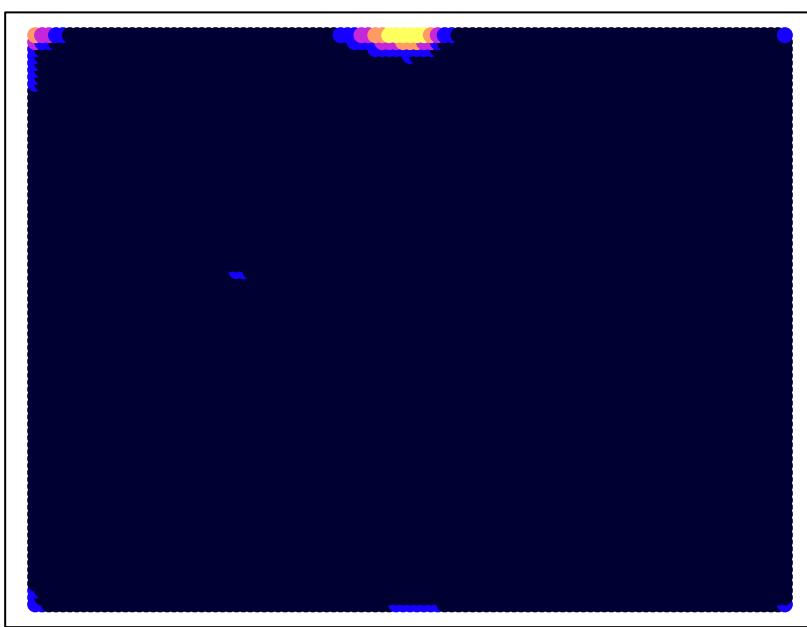
Kriging en Bloque

```
if(length(which(ls() == "intBK.C1")) == 0) {  
  
  # Interpolacion por Kriging en Bloque (BK)  
  intBK.C1 <- krigge(formula = maizC1 ~ 1,  
                      locations = datosRindeMaiz[-c(NAs, outliersC1),],  
                      newdata = puntosPredecir,  
                      model = varMejorTeoC1,  
                      block = c(60,60))  
}  
  
# Predicho  
spplot(intBK.C1, zcol = "var1.pred")
```



- [2.294,5.096]
- (5.096,7.898]
- (7.898,10.7]
- (10.7,13.5]
- (13.5,16.3]

```
spplot(intBK.C1, zcol = "var1.var")
```

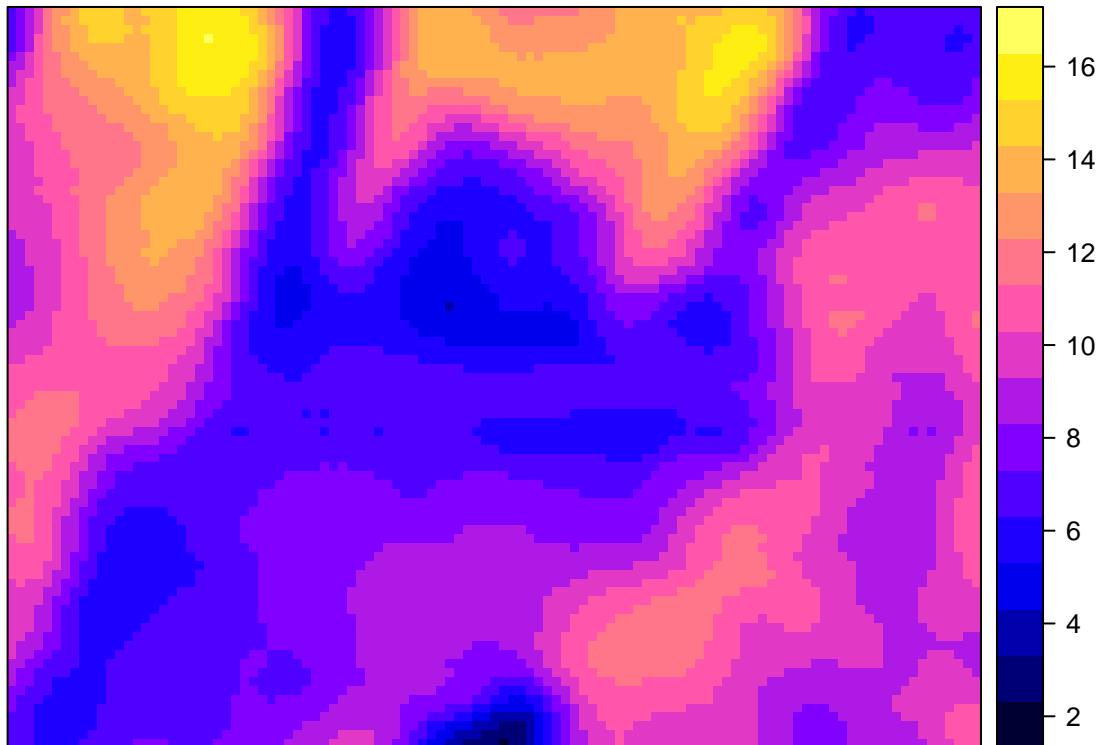


- [0.005127,0.07081]
- (0.07081,0.1365]
- (0.1365,0.2022]
- (0.2022,0.2679]
- (0.2679,0.3335]

```
# Visualizar como grilla
intBK.C1.grid <- SpatialPixelsDataFrame(points = intBK.C1,
                                         data = intBK.C1@data)

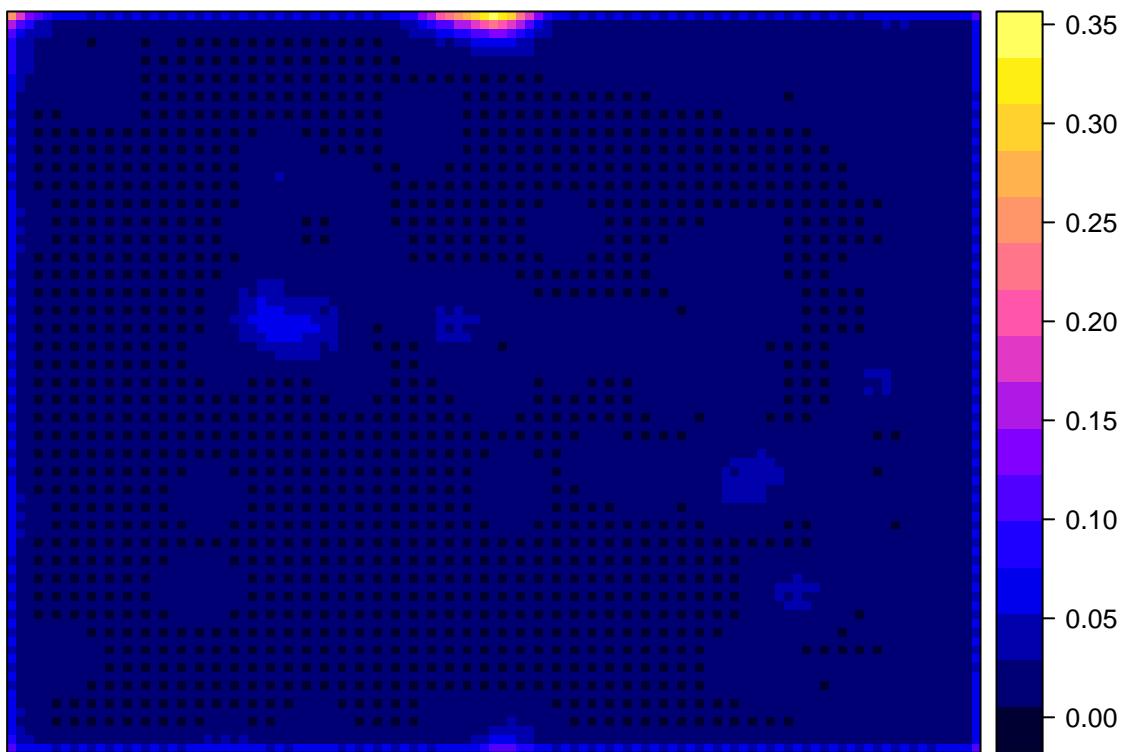
spplot(intBK.C1.grid, main = "Kriging en Bloque C1 - predicción", zcol = "var1.pred")
```

Kriging en Bloque C1 – predicción

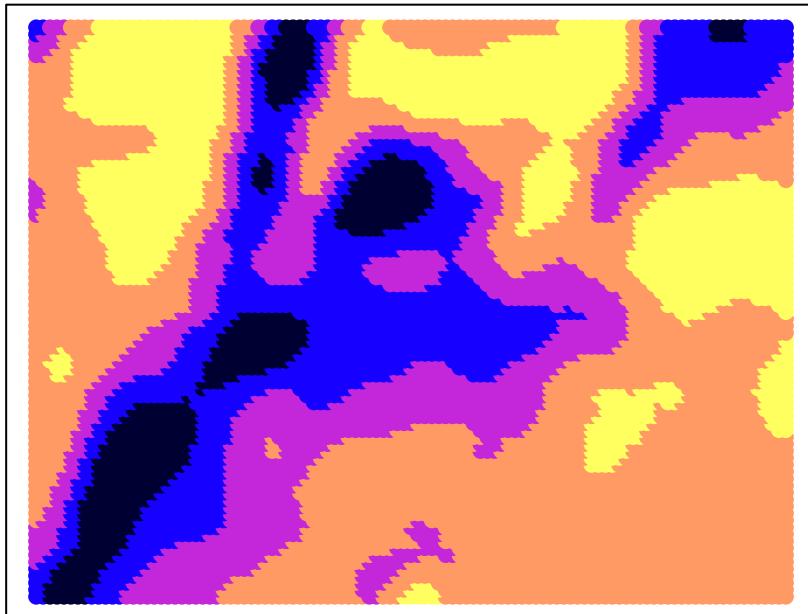


```
spplot(intBK.C1.grid, main = "Kriging en Bloque C1 - varianza", zcol = "var1.var")
```

Kriging en Bloque C1 – varianza

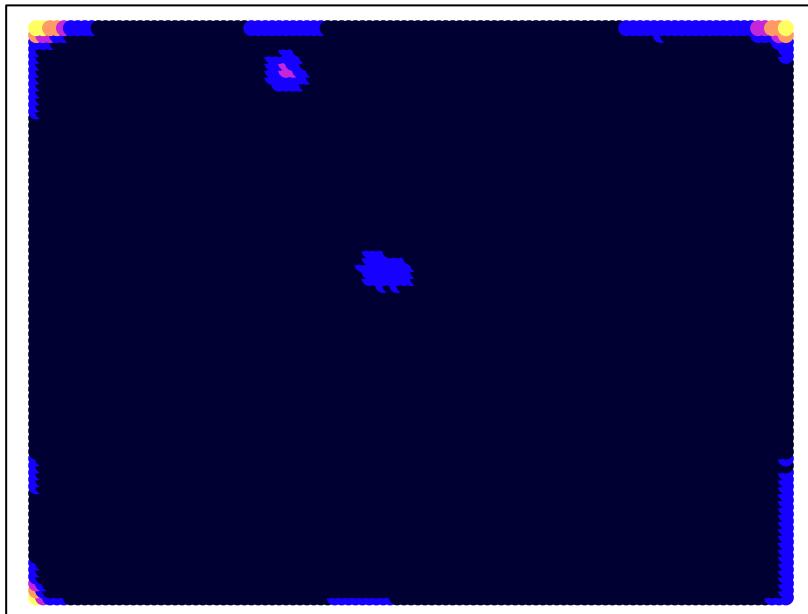


```
if(length(which(ls() == "intBK.C2")) == 0) {  
  
  # Interpolacion por Kriging en Bloque (BK)  
  intBK.C2 <- krigie(formula = maizC2 ~ 1,  
                      locations = datosRindeMaiz[-outliersC2,],  
                      newdata = puntosPredecir,  
                      model = varMejorTeoC2,  
                      block = c(60,60))  
}  
  
# Predicho  
spplot(intBK.C2, zcol = "var1.pred")
```



- [3.268,4.716]
- (4.716,6.165]
- (6.165,7.613]
- (7.613,9.061]
- (9.061,10.51]

```
spplot(intBK.C2, zcol = "var1.var")
```

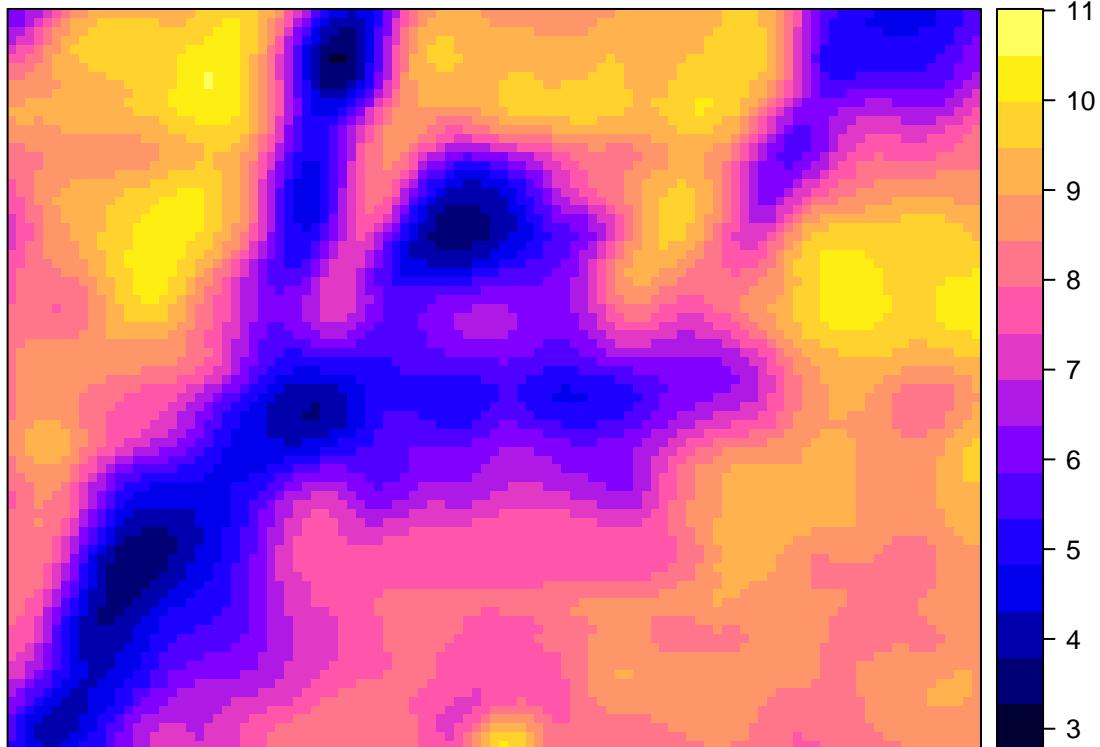


- [0.003374,0.035]
- (0.035,0.06664]
- (0.06664,0.09827]
- (0.09827,0.1299]
- (0.1299,0.1615]

```
# Visualizar como grilla
intBK.C2.grid <- SpatialPixelsDataFrame(points = intBK.C2,
                                         data = intBK.C2@data)

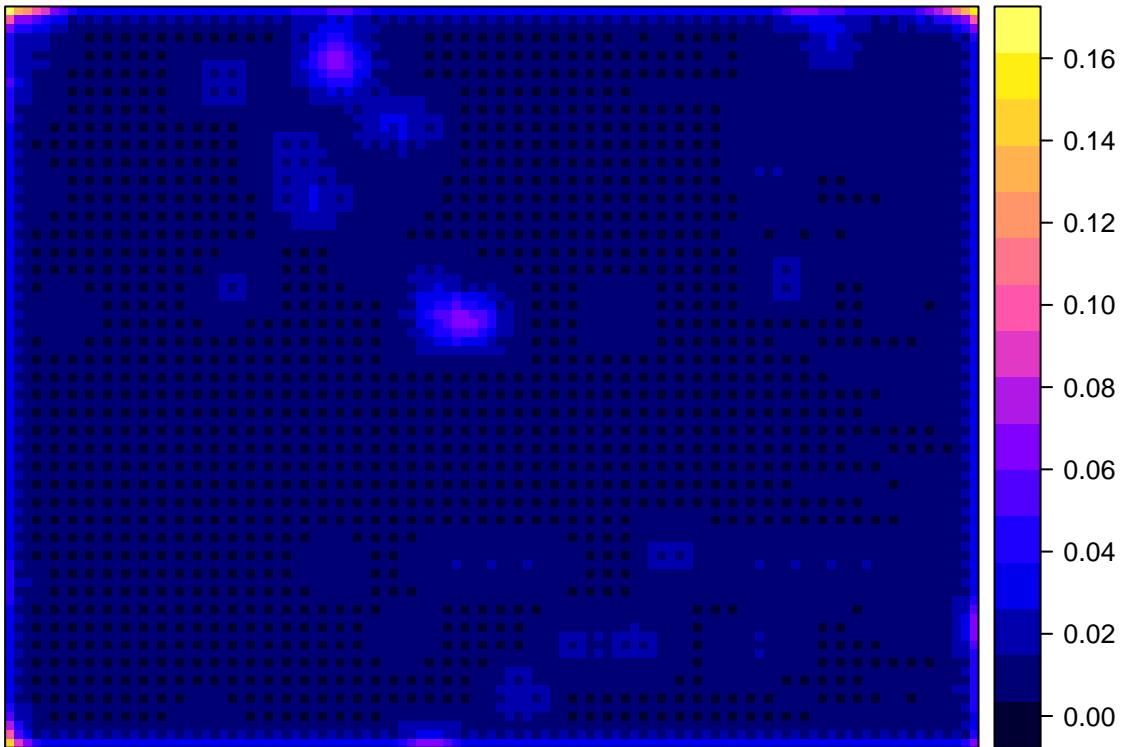
spplot(intBK.C2.grid, main = "Kriging en Bloque C2 - predicción", zcol = "var1.pred")
```

Kriging en Bloque C2 – predicción



```
spplot(intBK.C2.grid, main = "Kriging en Bloque C2 - varianza", zcol = "var1.var")
```

Kriging en Bloque C2 – varianza



Respuesta 5

- 5) Realice un procedimiento de validación cruzada para el método de interpolación puntual para las campañas analizadas en el punto anterior. Presente resultados sumarios respecto del error de predicción para toda el área estudiada.

Validación cruzada

Se observó que el valor de predicción global ($RMSE$) fue mejor para la C2 que para la C1. Para C1 se observaron valores de rendimiento bajos ($< 5t/ha$) que presentan diferencias importantes con los valores esperados. Sin embargo, se observó en general una buena predicción para las dos campañas como resumen algunos indicadores según el valor que se espera (ver tabla que se imprime más abajo).

```
# Validación cruzada mediante partición de los datos en 1000 partes.  
# Las predicciones estarán basadas en las restantes  $N-1$  partes para un conjunto  
# de observaciones de la parte  $N$ .  
  
if(length(which(ls() == "vcOK.C1")) == 0) {  
  
  vcOK.C1 <- krig.cv(formula = maizC1~1,  
                      model = varMejorTeoC1,  
                      locations = datosRindeMaiz[-c(NAs,outliersC1),],  
                      nfold = 1000)  
}  
  
if(length(which(ls() == "vcOK.C2")) == 0) {
```

```

vcOK.C2 <- krige.cv(formula = maizC2~1,
                      model = varMejorTeoC2,
                      locations = datosRindeMaiz[-outliersC2,],
                      nfold = 1000)
}

# Analizar validación cruzada
par(mfrow = c(2,2))

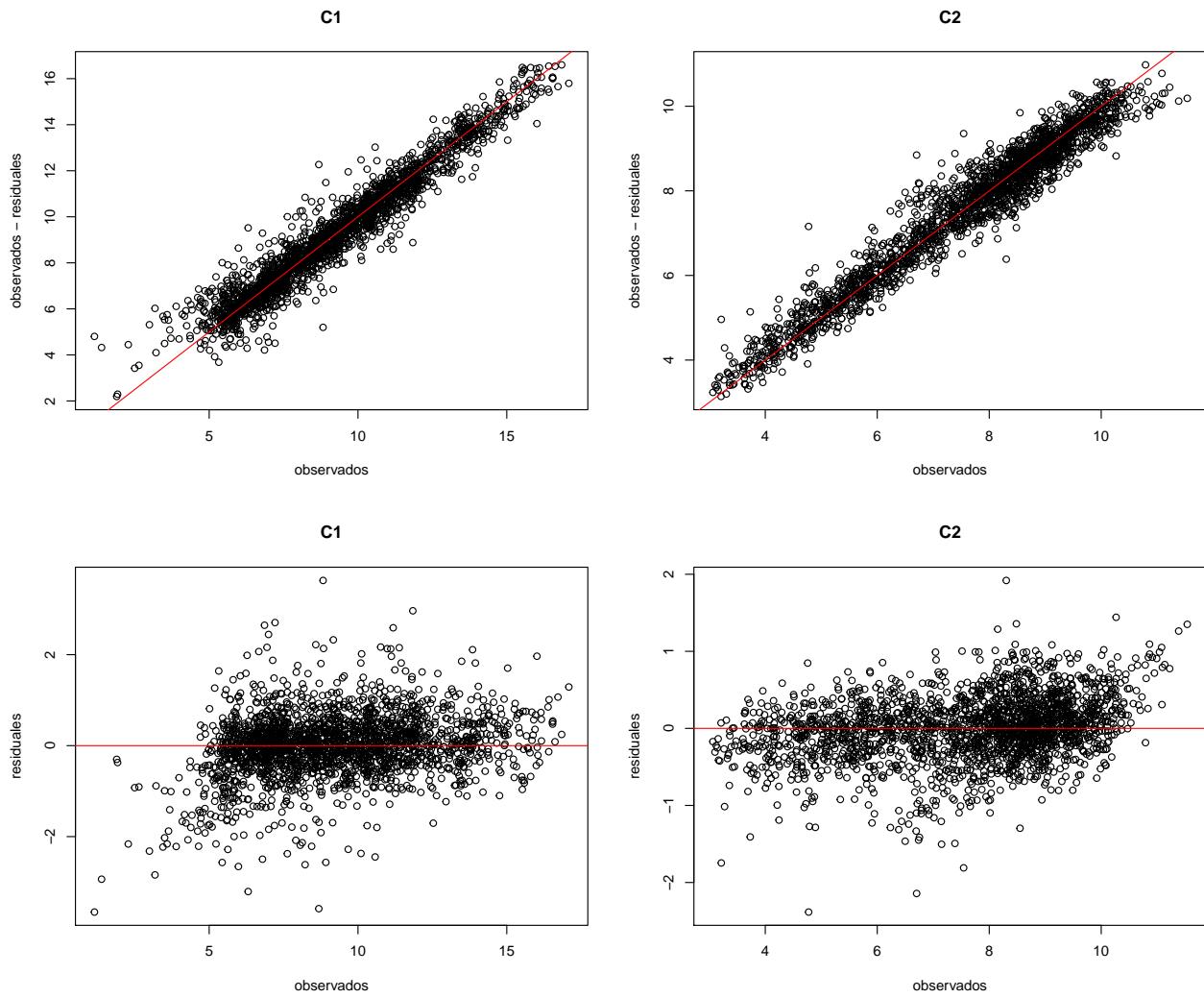
plot(x = vcOK.C1$observed,
      y = vcOK.C1$observed - vcOK.C1$residual,
      xlab = "observados",
      ylab = "observados - residuales",
      main = "C1")
lines(x = 0:20, y = 0:20, col = "red")

plot(x = vcOK.C2$observed,
      y = vcOK.C2$observed - vcOK.C2$residual,
      xlab = "observados",
      ylab = "observados - residuales",
      main = "C2")
lines(x = 0:20, y = 0:20, col = "red")

plot(x = vcOK.C1$observed,
      y = vcOK.C1$residual,
      xlab = "observados",
      ylab = "residuales",
      main = "C1")
lines(x = 0:19, y = rep(0,20), col = "red")

plot(x = vcOK.C2$observed,
      y = vcOK.C2$residual,
      xlab = "observados",
      ylab = "residuales",
      main = "C2")
lines(x = 0:19, y = rep(0,20), col = "red")

```



```

par(mfrow = c(1,1))

# Calcular RMSE para toda el área
rmse <- function(error) {
  sqrt(mean(error^2))
}

# Observados - predicho
RMSE.C1 <- rmse(vcOK.C1$residual)
RMSE.C2 <- rmse(vcOK.C2$residual)

# Tabla de resultados sumarios
vcResumen <- data.frame("Indicador" = c("Error.medio",
                                           "Error.medio.cuadrático",
                                           "Error.medio.cuadrático.normalizado",
                                           "Correlación.obsVSpred",
                                           "Correlación.predVSpresiduales",
                                           "RMSE"),
                           "ValorC1" = c(mean(vcOK.C1$residual),
                                         mean(vcOK.C1$residual^2),
                                         mean(vcOK.C1$zscore^2),
                                         mean(vcOK.C1$pred),
                                         mean(vcOK.C1$obs),
                                         RMSE.C1),
                           "ValorC2" = c(mean(vcOK.C2$residual),
                                         mean(vcOK.C2$residual^2),
                                         mean(vcOK.C2$zscore^2),
                                         mean(vcOK.C2$pred),
                                         mean(vcOK.C2$obs),
                                         RMSE.C2))
  
```

```

cor(vcOK.C1$observed, vcOK.C1$observed - vcOK.C1$residual),
cor(vcOK.C1$observed - vcOK.C1$residual, vcOK.C1$residual),
RMSE.C1),
"ValorC2" = c(mean(vcOK.C2$residual),
mean(vcOK.C2$residual^2),
mean(vcOK.C2$zscore^2),
cor(vcOK.C2$observed, vcOK.C2$observed - vcOK.C2$residual),
cor(vcOK.C2$observed - vcOK.C2$residual, vcOK.C2$residual),
RMSE.C2),
"Ideal" = c("0","0","1","1", "0", "0"))

# Ver
(vcResumen)

##          Indicador      ValorC1      ValorC2 Ideal
## 1        Error.medio -0.0004023642 -0.001589905 0
## 2  Error.medio.cuadrático  0.4812031353  0.172647943 0
## 3 Error.medio.cuadrático.normalizado  1.7405002999  0.845370420 1
## 4       Correlación.obsVSpred  0.9665152478  0.972647330 1
## 5 Correlación.predVSresiduales -0.0171878934  0.063189291 0
## 6             RMSE    0.6936880677  0.415509257 0

```

Respuesta 6

- 6) A partir de los modelos lineales regionalizados definidos en el punto 4, realice 300 simulaciones del proceso estocástico mediante kriging puntual ordinario para cada campaña. Asumiendo que ambas campañas fueron similares en cuanto al régimen de lluvias y la dinámica del acuífero elabore un mapa probabilístico que muestre la probabilidad de obtener rindes mayores a 9500 kg.ha⁻¹ integrando la información de ambas campañas. Justifique todas las decisiones necesarias para elaborar este mapa. ¿Cuál es la superficie de este lote que tiene una probabilidad igual o mayor a 0.8 de alcanzar o superar el rinde umbral establecido?

Simulaciones del proceso estocástico

Nota: Dado que para cada nodo visitado del set de datos el *algoritmo de simulación secuencial* simula un valor y este valor es agregado a los datos, el tiempo computacional requerido para realizar 300 simulaciones utilizando por defecto todos los puntos para cada nodo visitado es muy grande. Por eso se restringió a un radio de vecinos dado por todos los puntos encontrados a una distancia menor a 120m.

```

# Simulación por Kriging Ordinario (KO)

if(length(which(ls() == "KOSim.C1")) == 0) {

KOSim.C1 <- krige(formula = maizC1 ~ 1,
                   locations = datosRindeMaiz[-c(NAs,outliersC1),],
                   newdata = puntosPredecir,
                   model = varMejorTeoC1,
                   nsim = 300, maxdist = 120)
}

if(length(which(ls() == "KOSim.C2")) == 0) {

KOSim.C2 <- krige(formula = maizC2 ~ 1,

```

```

    locations = datosRindeMaiz[-outliersC2,],
    newdata = puntosPredecir,
    model = varMejorTeoC2,
    nsim = 300, maxdist = 120)
}

```

Cálculo de probabilidad

```

if(length(which(ls() == "probRinde")) == 0) {

# Probabilidad de obtener rindes mayores a 9500 kg.ha-1
probRinde <- numeric()
nroCeldas <- length(KOSim.C1$sim1)

# Recorro los datos simulados
for(i in 1:nroCeldas) {

    # Vectores
    C1Mayor <- length(which(KOSim.C1@data[i,] > 9.5))
    C2Mayor <- length(which(KOSim.C2@data[i,] > 9.5))

    # Condiciones
    ifelse(test = C1Mayor != 0, yes = probC1 <- C1Mayor/300, no = probC1 <- 0)
    ifelse(test = C2Mayor != 0, yes = probC2 <- C2Mayor/300, no = probC2 <- 0)

    # Calcular probabilidad
    probRinde[i] <- (probC1 + probC2) / 2

}

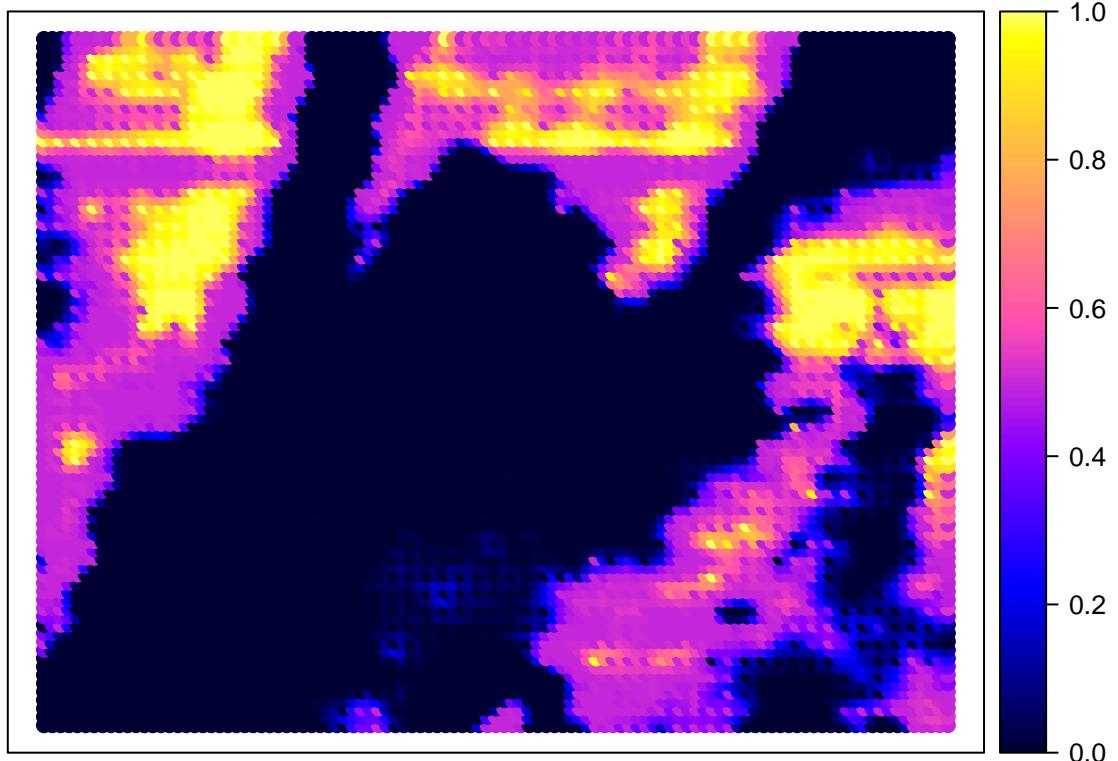
}

# Crear objeto espacial
probRindeMayor9500 <- SpatialPointsDataFrame(coords = KOSim.C1@coords,
                                                data = data.frame("ID" = 1:nrow(KOSim.C1@coords),
                                                                  "Prob" = probRinde))

# Plot
spplot(probRindeMayor9500, zcol = "Prob", colorkey = TRUE,
       main = "Probabilidad Rindes > 9500 kg.ha-1")

```

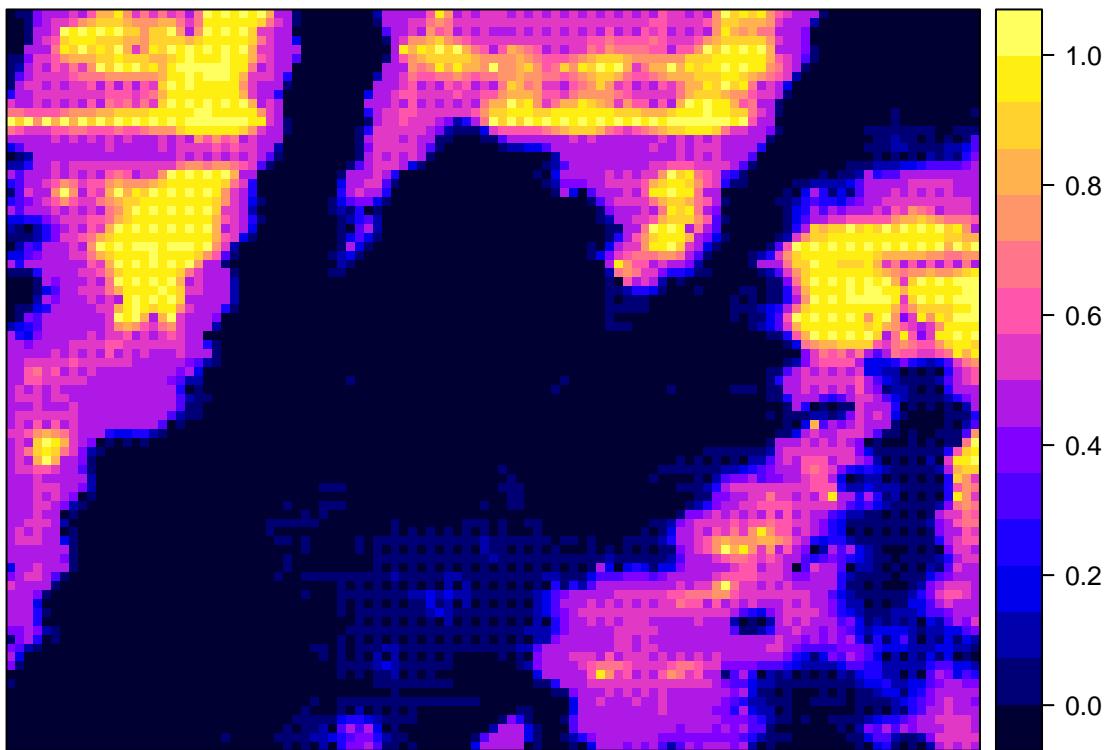
Probabilidad Rindes > 9500 kg.ha⁻¹



```
# Visualizar como grilla
probRindeMayor9500.grid <- SpatialPixelsDataFrame(points = puntosPredecir,
                                                    data = probRindeMayor9500@data)

spplot(probRindeMayor9500.grid, main = "Probabilidad Rindes > 9500 kg.ha-1",
       zcol = "Prob", colorkey = TRUE)
```

Probabilidad Rindes > 9500 kg.ha⁻¹



Cálculo de superficie

La superficie estimada del lote que tiene una probabilidad igual o mayor a 0.8 de alcanzar o superar el rinde umbral establecido fue de 88600m². Equivale a casi el 10% (9.79%) de la superficie total del lote.

```
# Consulto grilla creada anteriormente
mayorProbUmbral <- probRindeMayor9500.grid[which(probRindeMayor9500.grid$Prob>=0.8),]

# Tamaño de grilla:
mayorProbUmbral@grid@cellsize # 10 metros por 10 metros

## x1 x2
## 10 10

# Cacular superficie
areaCelda <- 10*10
cantidadCeldasCumplenCondicion <- length(mayorProbUmbral$Prob)
areaTotal <- cantidadCeldasCumplenCondicion * areaCelda
paste("Área total que cumple condición: ",
      areaTotal, " m2",
      sep = "")

## [1] "Área total que cumple condición: 88600 m2"

# Calcular porcentaje del lote
areaTotalLote <- length(probRindeMayor9500.grid$ID) * areaCelda
porcentajeAreaTotalCeldasCumplenCondicion <- (areaTotal/areaTotalLote)*100
paste("Porcentaje del Área total del lote que cumple condición: ",
```

```

round(porcentajeAreaTotalCeldasCumplenCondicion, digits = 2), " %",
sep = "")

## [1] "Porcentaje del Área total del lote que cumple condición: 9.79 %"

# Plot
spplot(mayorProbUmbbral, zcol = "Prob", colorkey = TRUE,
main = "Probabilidad > 0.8")

```

Probabilidad > 0.8

