Evaluación MDTE

Opción 2: Técnicas de krigeado sobre datos *meuse*

Inmaculada Perea Fernández

Junio 2017

Aplicar técnicas de krigeado a una de las variables logaritmo de la concentración de cadmio, cobre o plomo del conjunto de datos "meuse", realizando la predicción sobre el conjunto pixelado "meuse.grid".

Carga de librerías necesarias

```
if (!require('sp')) install.packages('sp'); library('sp')
if (!require('lattice')) install.packages('lattice'); library('lattice')
if (!require('xts')) install.packages('xts'); library('xts')
if (!require('gstat')) install.packages('gstat'); library('gstat')
```

1 Descripción de la variable (resumen y representaciones gráficas)

1.1 Carga de datos *meuse*

```
data(meuse)
dim(meuse)
```

```
## [1] 155  14
```

```
names(meuse)
```

```
##  [1] "x"       "y"       "cadmium" "copper"  "lead"    "zinc"    "elev"
##  [8] "dist"    "om"      "ffreq"   "soil"    "lime"    "landuse" "dist.m"
```

```
str(meuse)
```

```
## 'data.frame':    155 obs. of  14 variables:
##  $ x      : num  181072 181025 181165 181298 181307 ...
##  $ y      : num  333611 333558 333537 333484 333330 ...
##  $ cadmium: num  11.7 8.6 6.5 2.6 2.8 3 3.2 2.8 2.4 1.6 ...
##  $ copper : num  85 81 68 81 48 61 31 29 37 24 ...
##  $ lead   : num  299 277 199 116 117 137 132 150 133 80 ...
##  $ zinc   : num  1022 1141 640 257 269 ...
##  $ elev   : num  7.91 6.98 7.8 7.66 7.48 ...
##  $ dist   : num  0.00136 0.01222 0.10303 0.19009 0.27709 ...
##  $ om     : num  13.6 14 13 8 8.7 7.8 9.2 9.5 10.6 6.3 ...
##  $ ffreq  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ soil   : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 1 1 2 ...
##  $ lime   : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
##  $ landuse: Factor w/ 15 levels "Aa","Ab","Ag",..: 4 4 4 11 4 11 4 2 2 15 ...
##  $ dist.m : num  50 30 150 270 380 470 240 120 240 420 ...
```

```
head(meuse)
```

```
##        x      y cadmium copper lead zinc  elev       dist   om ffreq soil
## 1 181072 333611    11.7     85  299 1022 7.909 0.00135803 13.6     1    1
## 2 181025 333558     8.6     81  277 1141 6.983 0.01222430 14.0     1    1
```

```
## 3 181165 333537     6.5     68  199  640 7.800 0.10302900 13.0     1    1
## 4 181298 333484     2.6     81  116  257 7.655 0.19009400  8.0     1    2
## 5 181307 333330     2.8     48  117  269 7.480 0.27709000  8.7     1    2
## 6 181390 333260     3.0     61  137  281 7.791 0.36406700  7.8     1    2
##   lime landuse dist.m
## 1    1      Ah     50
## 2    1      Ah     30
## 3    1      Ah    150
## 4    0      Ga    270
## 5    0      Ah    380
## 6    0      Ga    470
```

El conjunto de datos *meuse* cuenta con 155 observaciones y 14 variables.

Este conjunto de datos proporciona ubicaciones y concentraciones de metales pesados en la capa superficial del suelo, junto con una serie de variables geofísicas, recogidos en una llanura de inundación del río Meuse.

Las observaciones están georreferenciadas en coordenadas UTM (x e y).

- **cadmium**: concentración de cadmio (ppm)
- **copper**: concentración de cobre (ppm)
- **lead**: concentración de plomo (ppm)
- **zinc**: concentración de zinc (ppm)
- **elev**: elevación sobre el nivel de lecho del rio (en metros)
- **dist**: distancia topométrica al grid o rejilla
- **om**: contenido de materia orgánica (%) del suelo
- **ffreq**: frecuencia de inundación (1=1 vez en 2 años; 2=1 vez en 10 a.; 3=1 vez en 50 años)
- **soil**: tipo de suelo (1-caliza; 2-arcilla pesada; 3-arcilla limosa)
- **lime**: clase de limo (0-ausente; 1-presente)
- **landuse**: uso de la parcela (diversas modalidades)
- **dist.m**: distancia al rio en metros, obtenida durante el trabajo de campo

1.2 Carga de los datos *meuse.grid*

```
data(meuse.grid)
dim(meuse.grid)
```

```
## [1] 3103    7
```

```
names(meuse.grid)
```

```
## [1] "x"      "y"      "part.a" "part.b" "dist"   "soil"   "ffreq"
```

```
str(meuse.grid)
```

```
## 'data.frame':    3103 obs. of  7 variables:
##  $ x     : num  181180 181140 181180 181220 181100 ...
##  $ y     : num  333740 333700 333700 333700 333660 ...
##  $ part.a: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ part.b: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ dist  : num  0 0 0.0122 0.0435 0 ...
##  $ soil  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ffreq : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(meuse.grid)
```

```
##        x      y part.a part.b      dist soil ffreq
```

```
## 1 181180 333740      1      0 0.0000000    1     1
## 2 181140 333700      1      0 0.0000000    1     1
## 3 181180 333700      1      0 0.0122243    1     1
## 4 181220 333700      1      0 0.0434678    1     1
## 5 181100 333660      1      0 0.0000000    1     1
## 6 181140 333660      1      0 0.0122243    1     1
```

El conjunto de datos *meuse.grid* cuenta con 3103 observaciones y 7 variables.

Las observaciones están georreferenciadas en coordenadas UTM (x e y). En cada localización se han recogido:

- **x**: Coordenada X
- **y**: Coordenada Y
- **dist**: distancia al borde de río Meuse, normalizada a [0,1]
- **ffreq**: freq.inundación (1=1 vez en 2 años; 2=1 vez en 10 a.; 3=1 vez en 50 años)
- **soil**: tipo de suelo (1-caliza; 2-arcilla pesada; 3-arcilla limosa)
- **part.a**: división arbitraria del área en dos zonas, Zona A
- **part.b**: división arbitraria del área en dos zonas, Zona B

1.3 Preparación de los datos

1.3.1 Datos *meuse*

```
coordinates(meuse) = ~x+y  # Asignación de coordenadas
class(meuse)
```

```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
names(meuse)
```

```
##  [1] "cadmium" "copper"  "lead"    "zinc"    "elev"    "dist"    "om"
##  [8] "ffreq"   "soil"    "lime"    "landuse" "dist.m"
```

1.3.2 Datos *meuse.grid*

```
coordinates(meuse.grid) = ~x+y  # Asignación de coordenadas
gridded(meuse.grid) = TRUE      # Determinación como "rejilla"
class(meuse.grid)
```

```
## [1] "SpatialPixelsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
names(meuse.grid)
```

```
## [1] "part.a" "part.b" "dist"   "soil"   "ffreq"
```

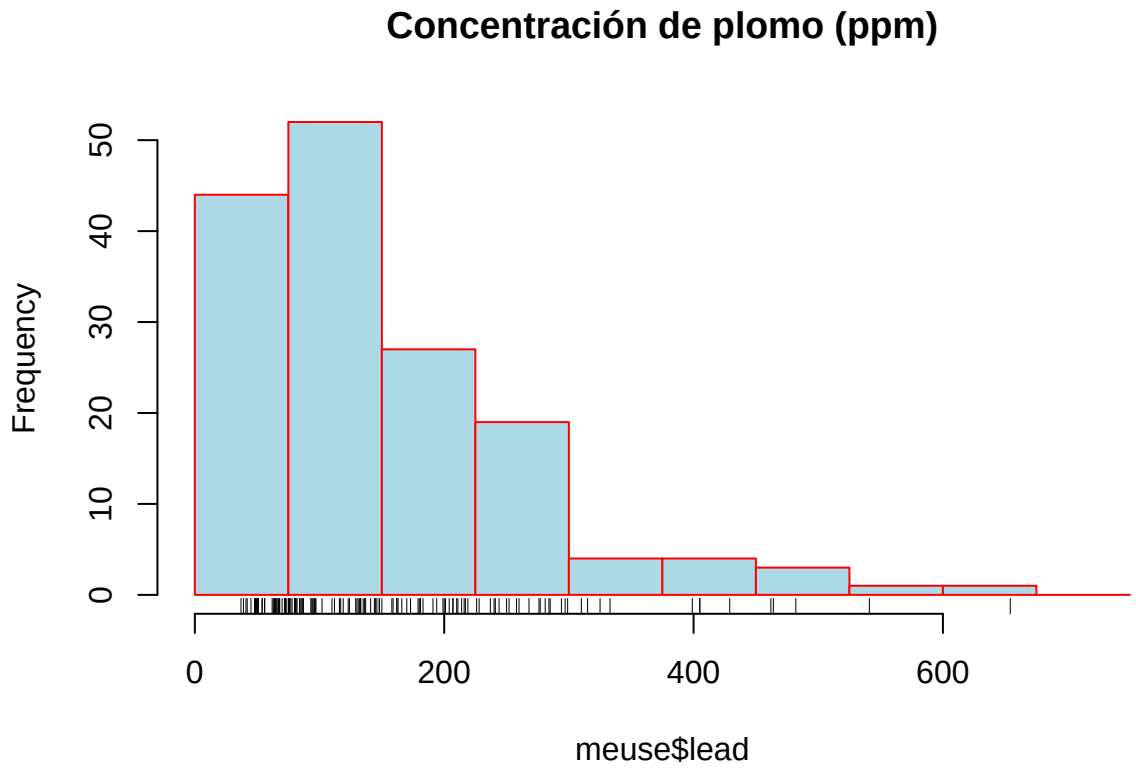
Después de la asignación de coordenadas el conjunto de datos deja de ser un dataframe para ser un dataframe de PUNTOS espaciales (puntos, no poligonos)

1.4 Representación gráfica

Nos centraremos en la variable concentración de **plomo (lead)**.

1.4.1 Histograma de la variable *lead*

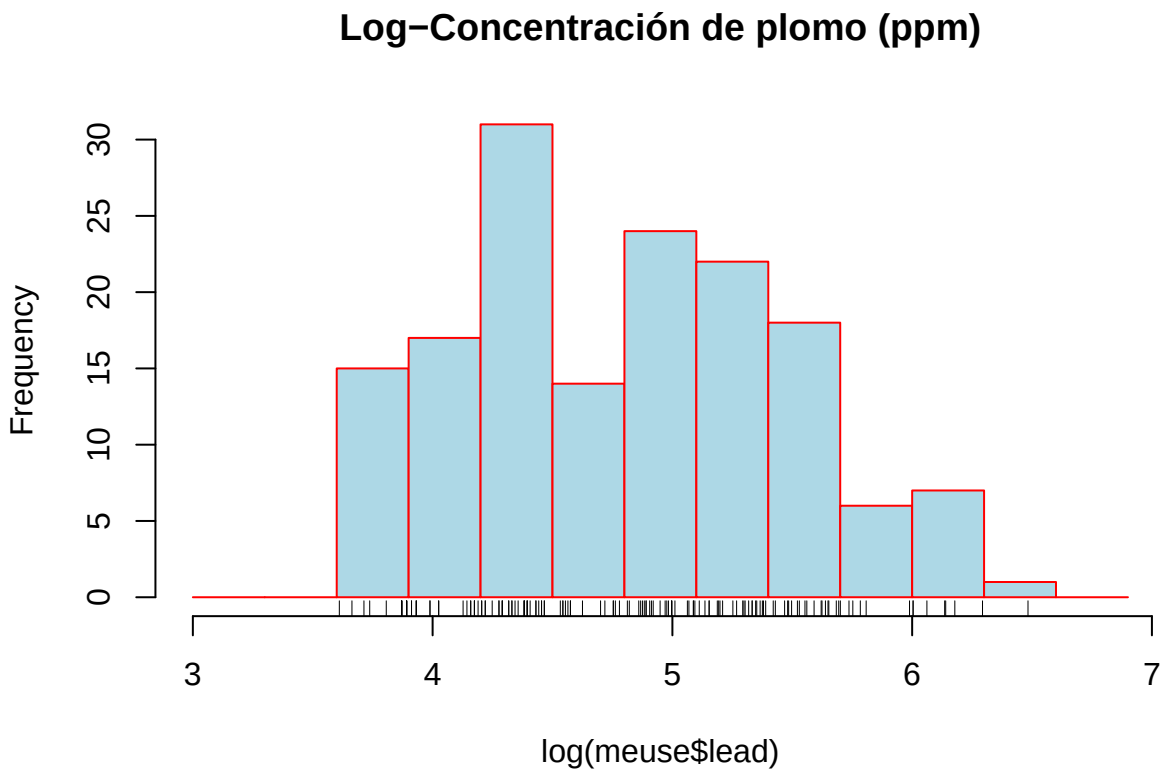
```
hist(meuse$lead,
     breaks = seq(0, 800, by = 75),
     col = "lightblue",
     border = "red",
     main = "Concentración de plomo (ppm)")

rug(meuse$lead)
```




```
# Buscando un comportamiento aproximado a la normal se utiliza la transformación logarítmica
summary(log(meuse$lead))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.611   4.284   4.812   4.807   5.333   6.483
```

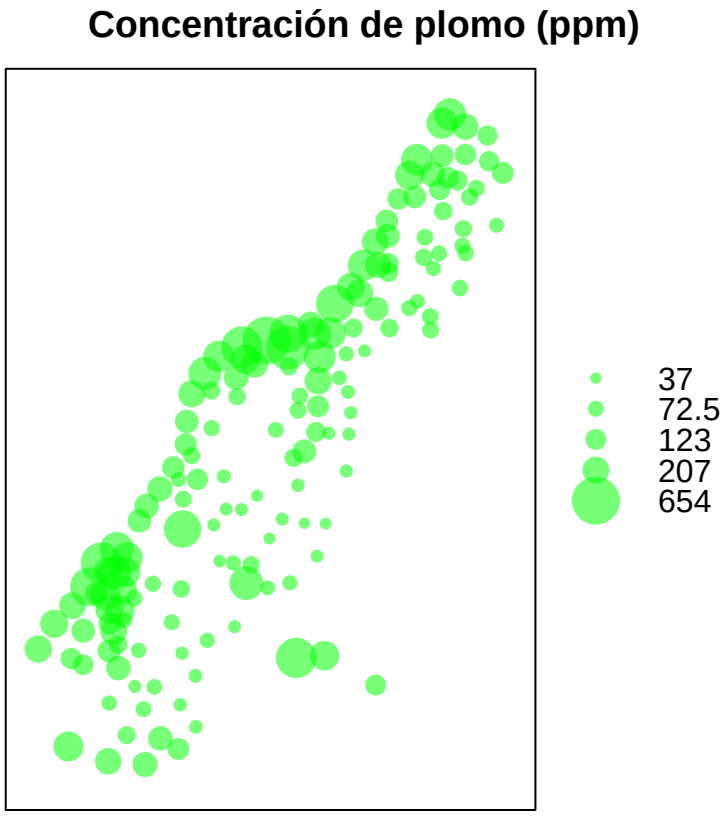
```
hist(log(meuse$lead),
     breaks = seq(3, 7, by = 0.3),
     col = "lightblue",
     border = "red",
     main = "Log-Concentración de plomo (ppm)")

rug(log(meuse$lead))
```



Con un comportamiento gaussiano no necesito tantas muestras para que funcione bien.

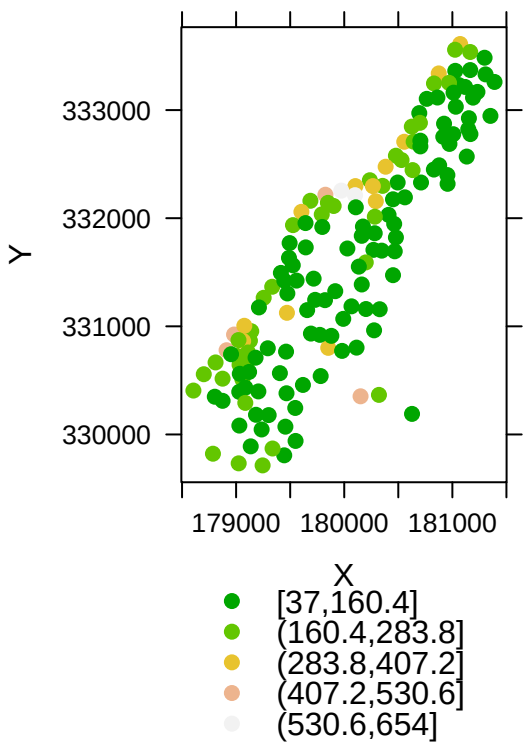
1.4.2 Gráfica de burbujas

```
bubble(meuse, c("lead"), col=c("#00ff0088", "#00ff0088"), main = "Concentración de plomo (ppm)")
```

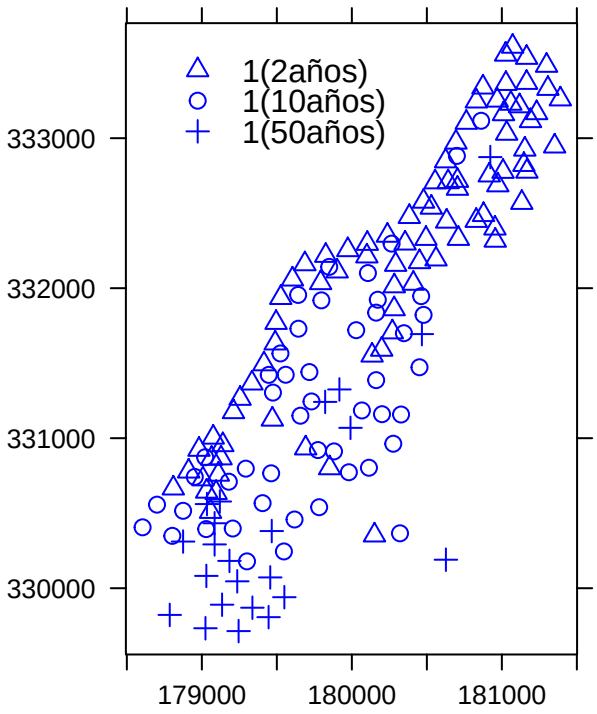


1.4.3 Gráficos de puntos/colores o caracteres

```
spplot(meuse["lead"],
       main="concentración de plomo",
       scales=list(draw=TRUE),
       xlab="X", ylab="Y",
       col.regions=terrain.colors(10))
```

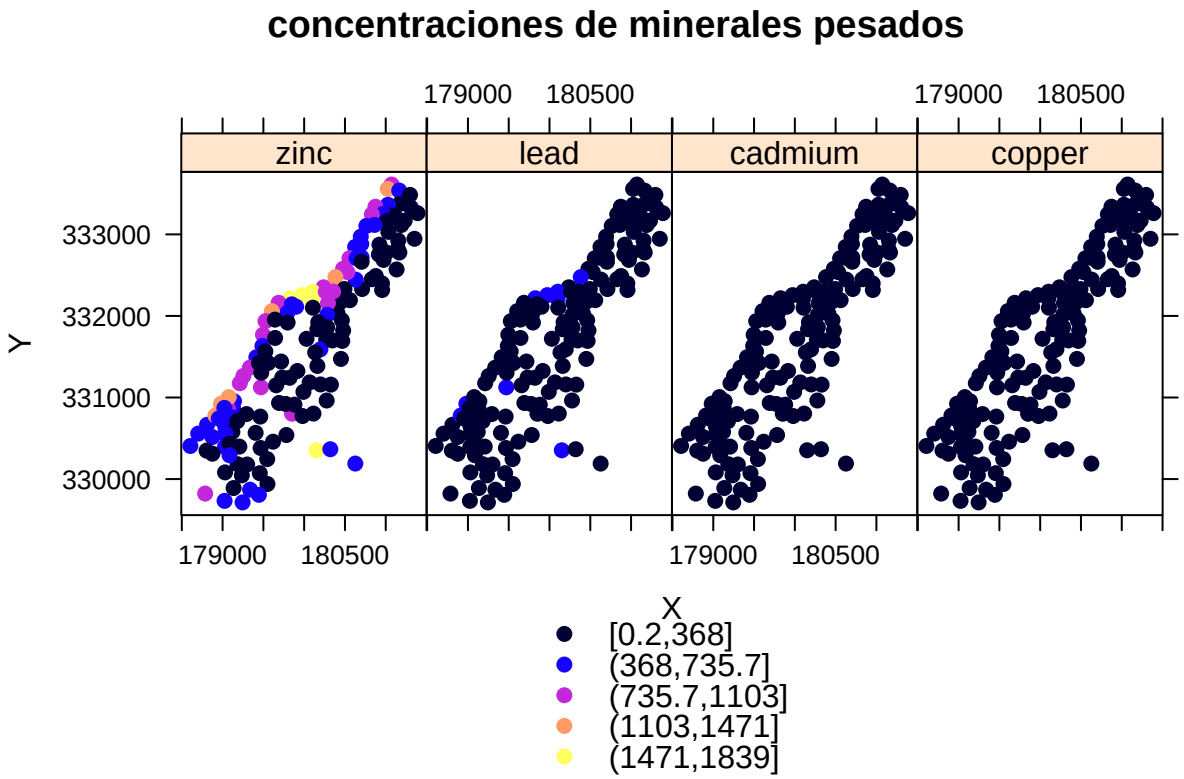



```
spplot(meuse, c("ffreq"),
       cex=1, pch=c(2,1,3),
       scales=list(draw=TRUE),
       legendEntries=c("1(2años)","1(10años)","1(50años)"),
       main=" Frecuencia de inundacion ",
       col.regions= "blue",
       key.space=list(x=0.1,y=.95,corner=c(0,1)))
```

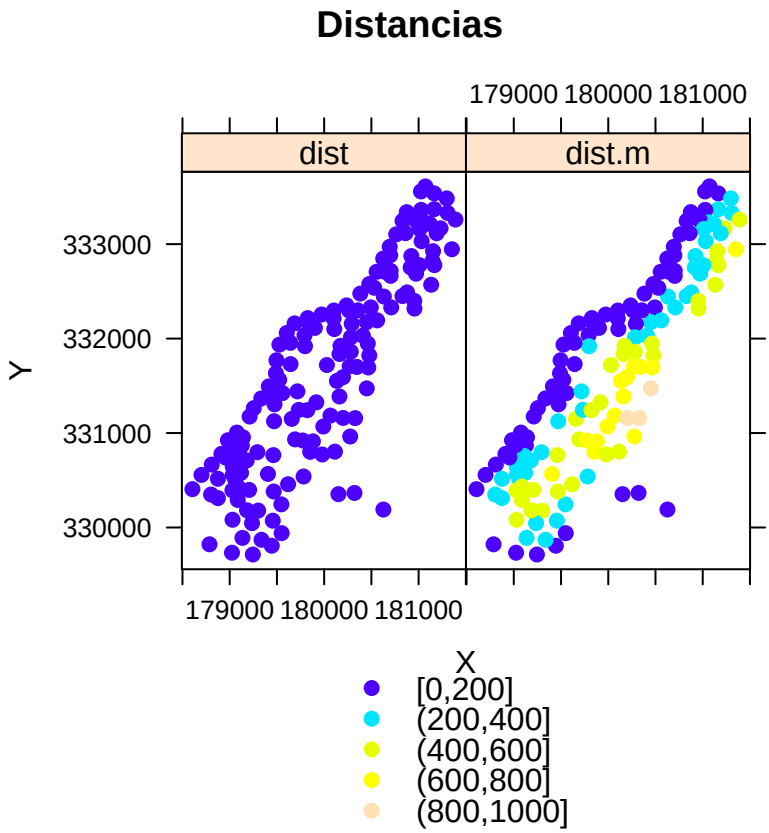


1.4.4 Gráficos múltiples

```
spplot(meuse, c("zinc","lead","cadmium","copper"),
       main="concentraciones de minerales pesados",
       scales=list(draw=TRUE),
       xlab="X", ylab="Y")
```

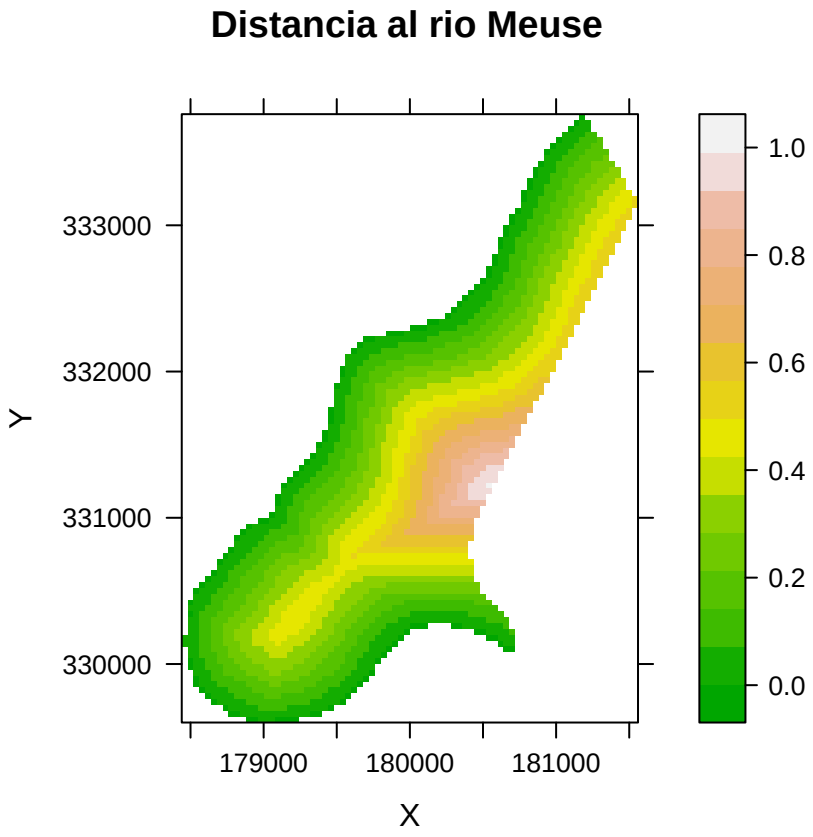



```
spplot(meuse, c("dist","dist.m"),main="Distancias",
       col.regions=topo.colors(6),
       scales=list(draw=TRUE),
       xlab="X", ylab="Y")
```



1.4.5 Plots para visualizar el fichero *meuse.grid*

```
spplot(meuse.grid, c("dist"),
       col.regions=terrain.colors(20),
       main="Distancia al rio Meuse",
       scales=list(draw=TRUE),
       xlab="X", ylab="Y")
```



2 Construcción del variograma muestral y ajuste a un modelo teórico de la variable objetivo

2.1 Construcción del variograma muestral

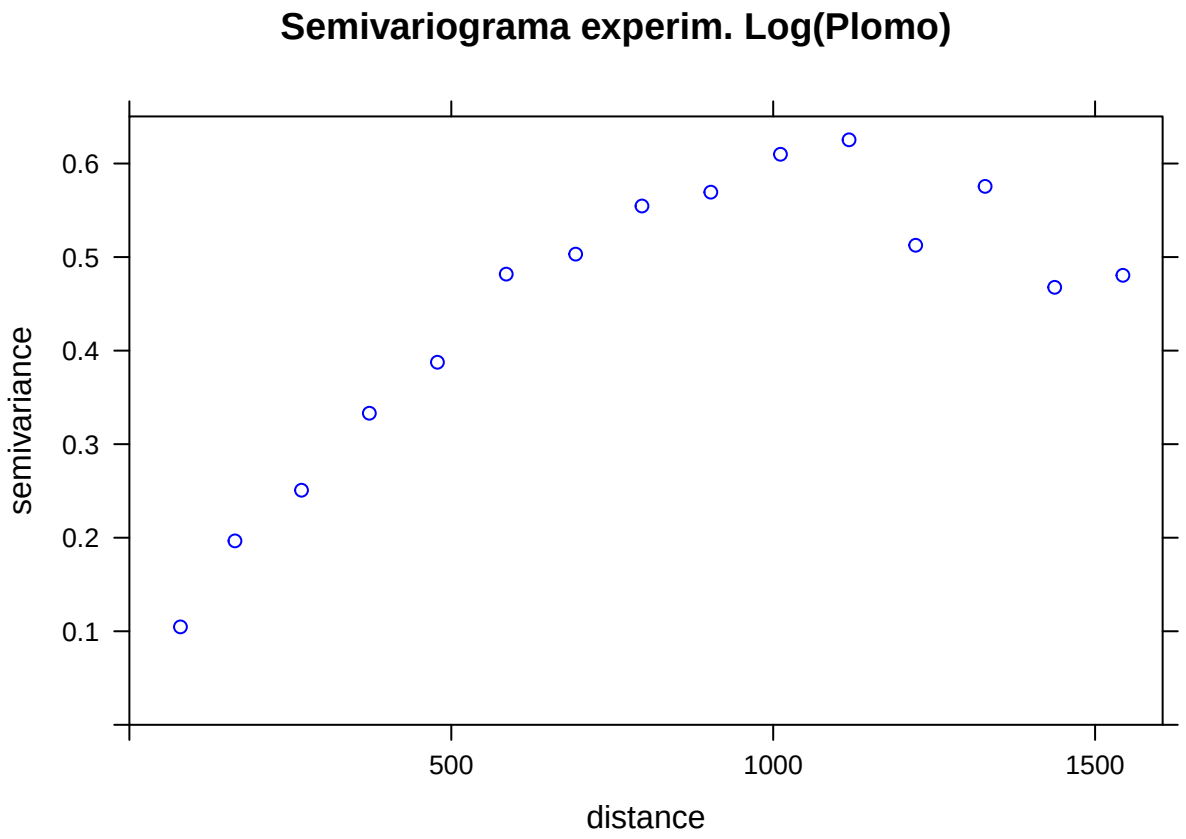
Calculo el variograma muestral de la variable *lead* con la funcion *variogram*

```
(lead.variogram = variogram(log(lead)~1,  # object: variable
                            meuse))       # conjunto de datos
```

```
##     np       dist     gamma dir.hor dir.ver   id
## 1   57   79.29244 0.1046520       0       0 var1
## 2  299  163.97367 0.1965929       0       0 var1
## 3  419  267.36483 0.2507668       0       0 var1
## 4  457  372.73542 0.3330690       0       0 var1
## 5  547  478.47670 0.3875716       0       0 var1
## 6  533  585.34058 0.4817750       0       0 var1
## 7  574  693.14526 0.5031432       0       0 var1
## 8  564  796.18365 0.5545787       0       0 var1
## 9  589  903.14650 0.5693882       0       0 var1
## 10 543 1011.29177 0.6098806       0       0 var1
## 11 500 1117.86235 0.6253271       0       0 var1
## 12 477 1221.32810 0.5126165       0       0 var1
## 13 452 1329.16407 0.5755737       0       0 var1
```

```
## 14 457 1437.25620 0.4676728       0       0 var1
## 15 415 1543.20248 0.4804887       0       0 var1
```

```
plot(lead.variogram, col="blue", main="Semivariograma experim. Log(Plomo)")
```



Se observa una aproximacion al efecto pepita.

2.2 Ajuste al modelo teórico

Utilizaremos la función *fit.variogram* para elegir de entre todos los modelos el mejor. Con la función *vgm* generaremos un modelo de variograma según el modelo indicado en el parámetro *model* (por defecto Sph)

Vamos a realizar ajustes con distintos modelos para elegir el que menor error cuadrático medio

```
cat(" Spherical = ", attributes(fit.variogram(lead.variogram, model=vgm(1, "Sph", 900, 1)))$SSErr, "\n"
"Pentaspherical = ", attributes(fit.variogram(lead.variogram, model=vgm(1, "Pen", 900, 1)))$SSErr, "\n"
"Gaussian = ", attributes(fit.variogram(lead.variogram, model=vgm(1, "Gau", 900, 1)))$SSErr, "\n",
"Circular = ", attributes(fit.variogram(lead.variogram, model=vgm(1, "Cir", 900, 1)))$SSErr, "\n",
"Exponential = ", attributes(fit.variogram(lead.variogram, model=vgm(1, "Exp", 900, 1)))$SSErr)
```

```
##  Spherical =  1.211742e-05
##  Pentaspherical =  1.263889e-05
##  Gaussian =  2.450349e-05
##  Circular =  1.232785e-05
##  Exponential =  2.051716e-05
```

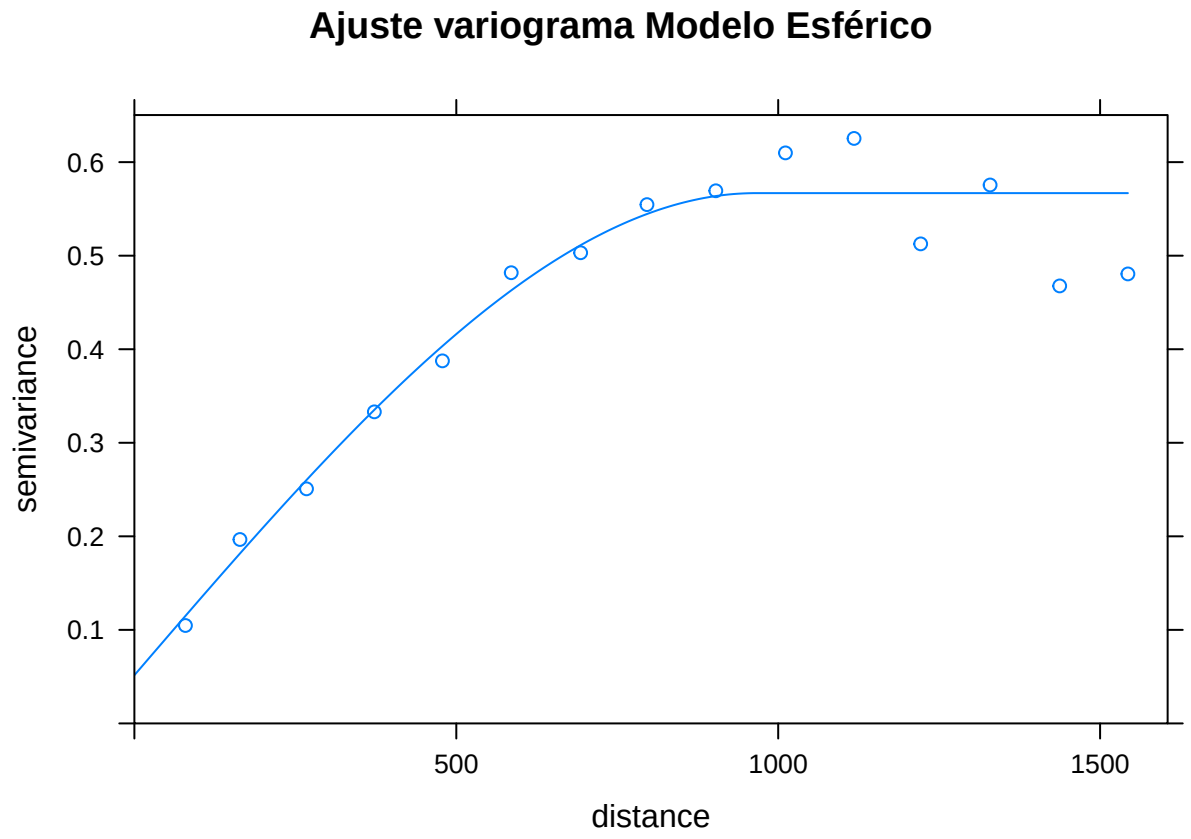
Según el anteior criterio se selecciona el modelo **Esférico**.

```
lead.fit = fit.variogram(lead.variogram,                    # variograma muestral
                         model = vgm(0.5, "Sph", 900, 0.1))  # valores iniciales del modelo
lead.fit
```

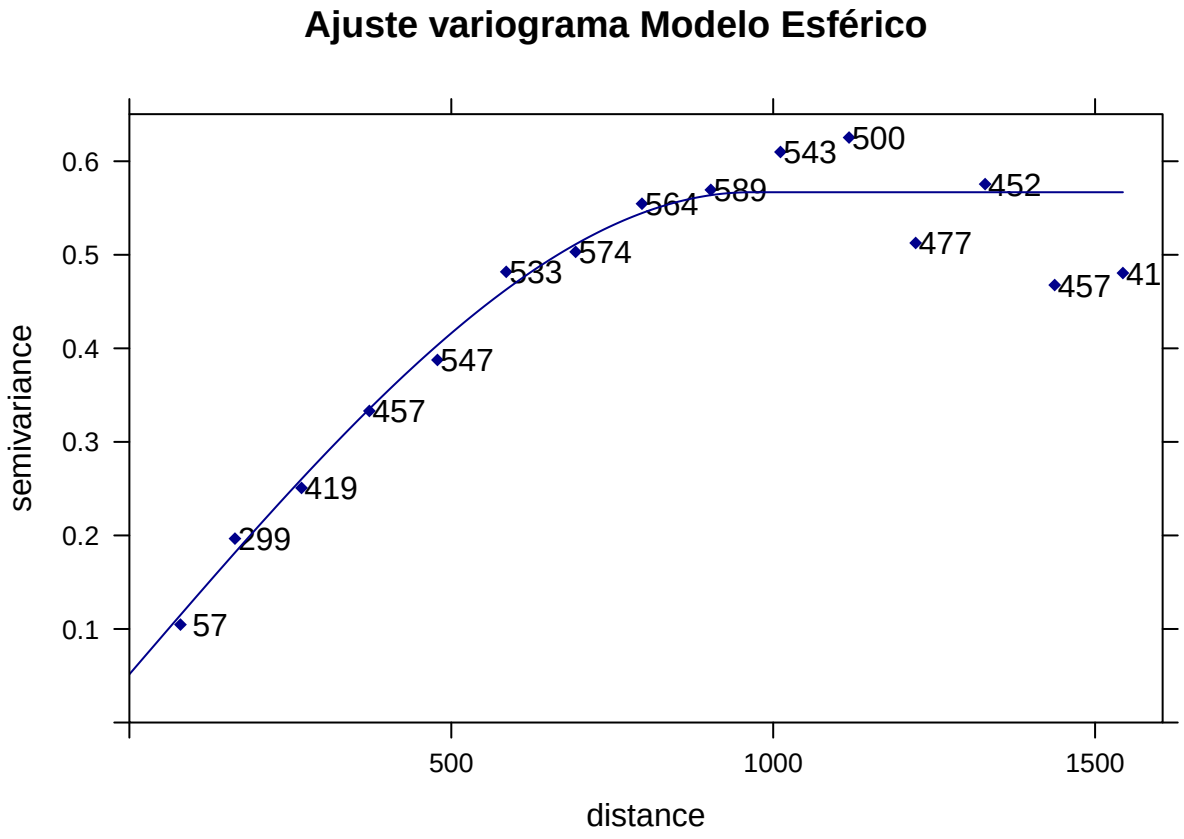
```
##   model      psill    range
## 1   Nug 0.05156304   0.0000
## 2   Sph 0.51530751 965.1558
```

Se ha creado el objeto *lead.fit* que contiene el modelo esférico estimado de la variable log(lead) con la siguiente información

```
plot(lead.variogram, lead.fit, main="Ajuste variograma Modelo Esférico")
```




```
print(plot(lead.variogram, plot.numbers = T, pch =18, col = "darkblue",
           model = lead.fit, main="Ajuste variograma Modelo Esférico"))
```



3 Kriging ordinario para la variable objetivo

3.1 Cálculo de las predicciones kriging ordinario

Utilizamos las funcion *krige* para realizar las predicciones *kriging* sobre los puntos definidos en el grid

```
lead.kriged = krige(log(lead)~1,       # formula
                    meuse,             # datos espaciales
                    meuse.grid,        # datos donde se van a hacer las predicciones
                    model = lead.fit)  # modelo
```

```
## [using ordinary kriging]
```

Se ha creado un objeto que se describe en las siguientes órdenes:

```
names(lead.kriged)
```

```
## [1] "var1.pred" "var1.var"
```

```
dim(lead.kriged)
```

```
## [1] 3103    2
```

```
lead.kriged$var1.pred[1:5]   # Predicción en los cinco primeros casos
```

```
## [1] 5.365832 5.447747 5.373107 5.297427 5.538664
```

```
lead.kriged$var1.var[1:5]     # Varianza de la Predicción en los cinco primeros casos
```

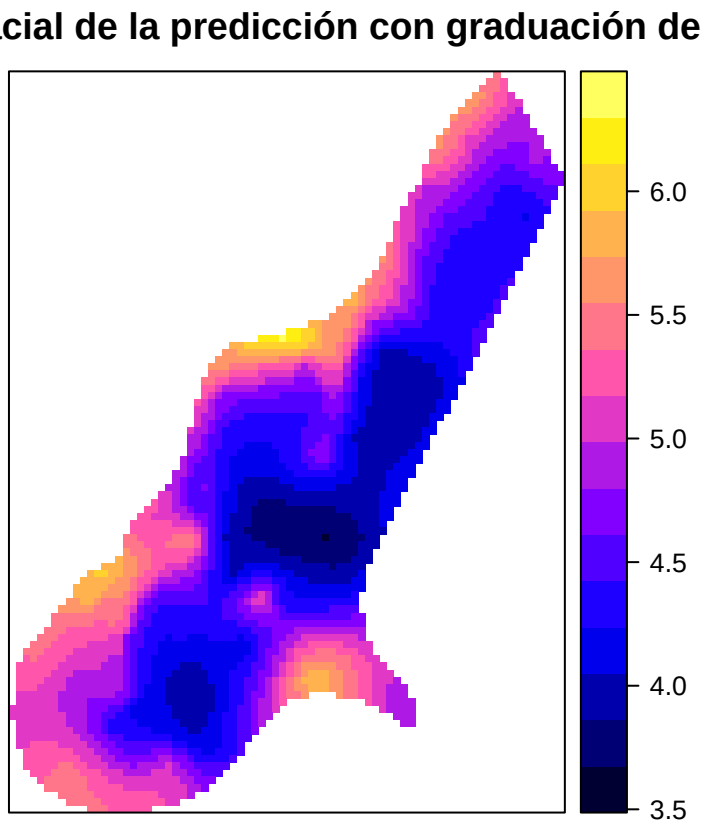
```
## [1] 0.2755230 0.2197892 0.2366098 0.2549536 0.1596112
```

Las zonas de la frontera superior son las zonas con mayor concentración de plomo. Las zonas mas alejadas del rio tienen menos concentración. Por tanto, parece que el origen de la contaminación es el rio.

3.2 Representación gráfica

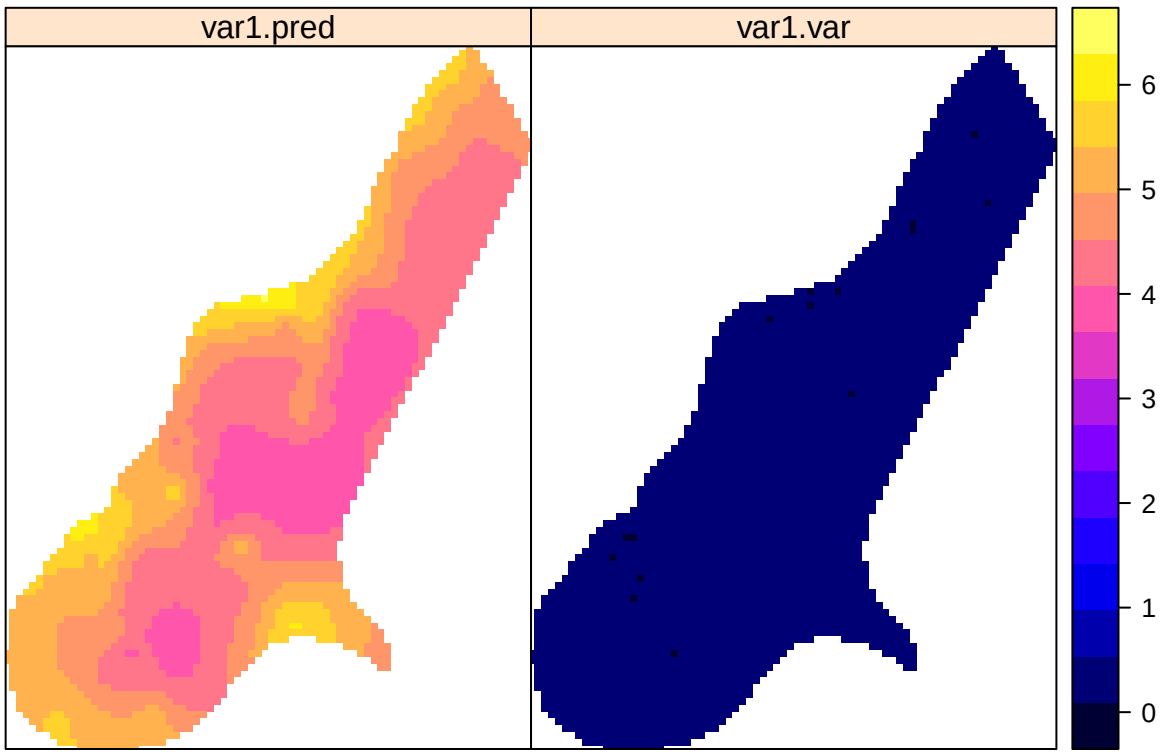
Plot espacial de la predicción con graduación de colores

```
spplot(lead.kriged["var1.pred"], main="Plot espacial de la predicción con graduación de colores" )
```



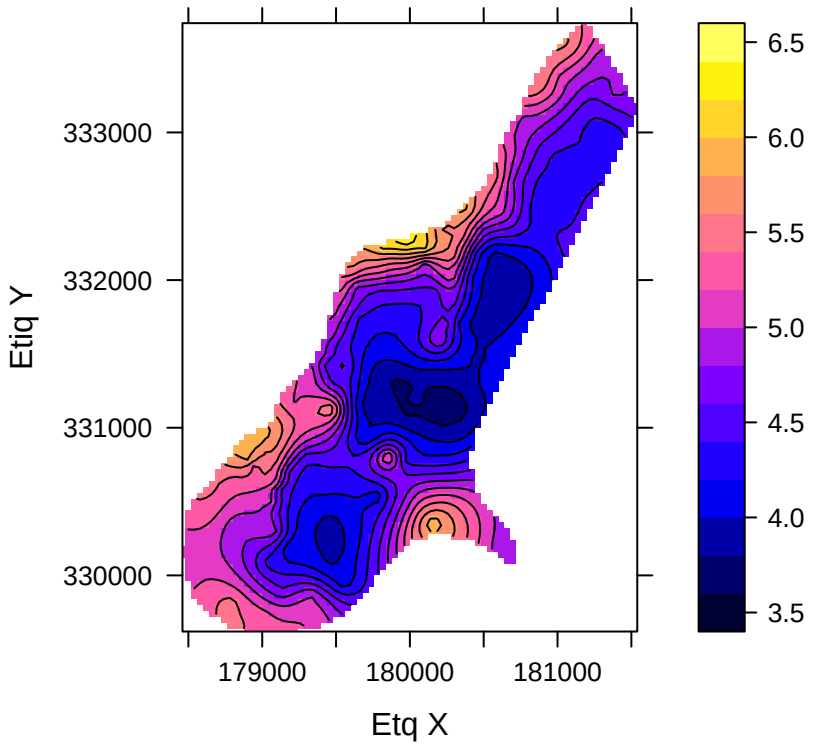
Plot espacial de la predicción y la varianza de la predicción con graduación de colores

```
spplot(lead.kriged, main="Plots espaciales de la predicción y la varianza de la predicción" )
```



Plot espacial de la predicción con líneas de contorno

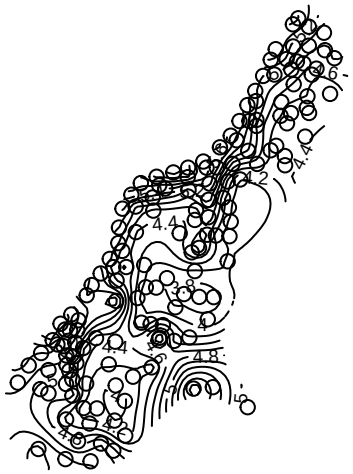
```
spplot(lead.kriged, zcol="var1.pred", pretty=T, contour=T, col.regions=bpy.colors(64),
       main="Plot espacial de la predicción con líneas de contorno",
       xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))
```



Representación de curva de nivel de las predicciones, incluyendo los puntos observados

```
contour(lead.kriged, main="Curva de nivel de las predicciones con las localizaciones muestrales")
points(coordinates(meuse))
```

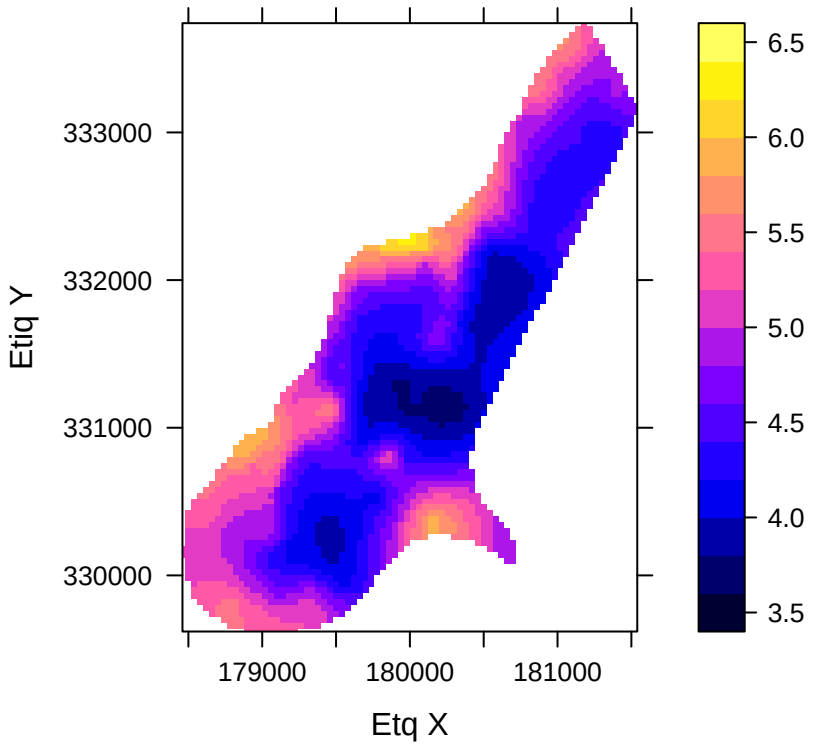
Curva de nivel de las predicciones con las localizaciones muestrale



Representación espacial de la predicción con graduación de colores

```
spplot(lead.kriged, zcol="var1.pred", pretty=T, col.regions=bpy.colors(64),
       main="Plot espacial de la predicción con graduación de colores",
       xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))
```

Plot espacial de la predicción con graduación de colores



4 Kriging universal para la variable objetivo

Se realizará el estudio con un polinomio de grado 1 (1,x,y) para la variable objetivo *lead* (plomo)

4.1 Ajuste lineal y determinación de residuos

En primer lugar, realizaremos la búsqueda de la tendencia lineal (en función de las coordenadas). Obtenemos el modelo de regresión para la variable original *lead* y su transformada log(lead) y determinamos cuál de ellos resulta más adecuado.

```
summary(lm(formula=lead ~ coordinates(meuse), data=meuse))
```

```
##
## Call:
## lm(formula = lead ~ coordinates(meuse), data = meuse)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -139.38  -62.15  -30.87   33.76  445.69
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.067e+03  2.625e+03  -1.930   0.0554 .
## coordinates(meuse)x -1.227e-01  2.124e-02  -5.779 4.11e-08 ***
```

```
## coordinates(meuse)y  8.236e-02  1.512e-02   5.446 2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 101.2 on 152 degrees of freedom
## Multiple R-squared:  0.1842, Adjusted R-squared:  0.1734
## F-statistic: 17.16 on 2 and 152 DF,  p-value: 1.913e-07
```

```
llead<-log(meuse$lead)
summary(lm(formula=llead ~ coordinates(meuse), data=meuse))
```

```
##
## Call:
## lm(formula = llead ~ coordinates(meuse), data = meuse)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98377 -0.40009 -0.05057  0.35965  2.22399
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -3.917e+01  1.500e+01  -2.612  0.00989 **
## coordinates(meuse)x -8.563e-04  1.213e-04  -7.058 5.61e-11 ***
## coordinates(meuse)y  5.974e-04  8.638e-05   6.916 1.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5781 on 152 degrees of freedom
## Multiple R-squared:  0.2572, Adjusted R-squared:  0.2475
## F-statistic: 26.32 on 2 and 152 DF,  p-value: 1.531e-10
```

Resulta más adecuado el modelo sobre log(lead), ya que presenta mayor R2.

A continuación se procede a estimar el variograma asociado a los residuos del modelo.

4.2 Estimación del variograma de los residuos MCO

Se crea el variograma experimental o muestral de los residuos (objeto "lead.res.variogram") y se compara con el semivariograma muestral obtenido para los datos originales, almacenado en el objeto "lead.variogram"

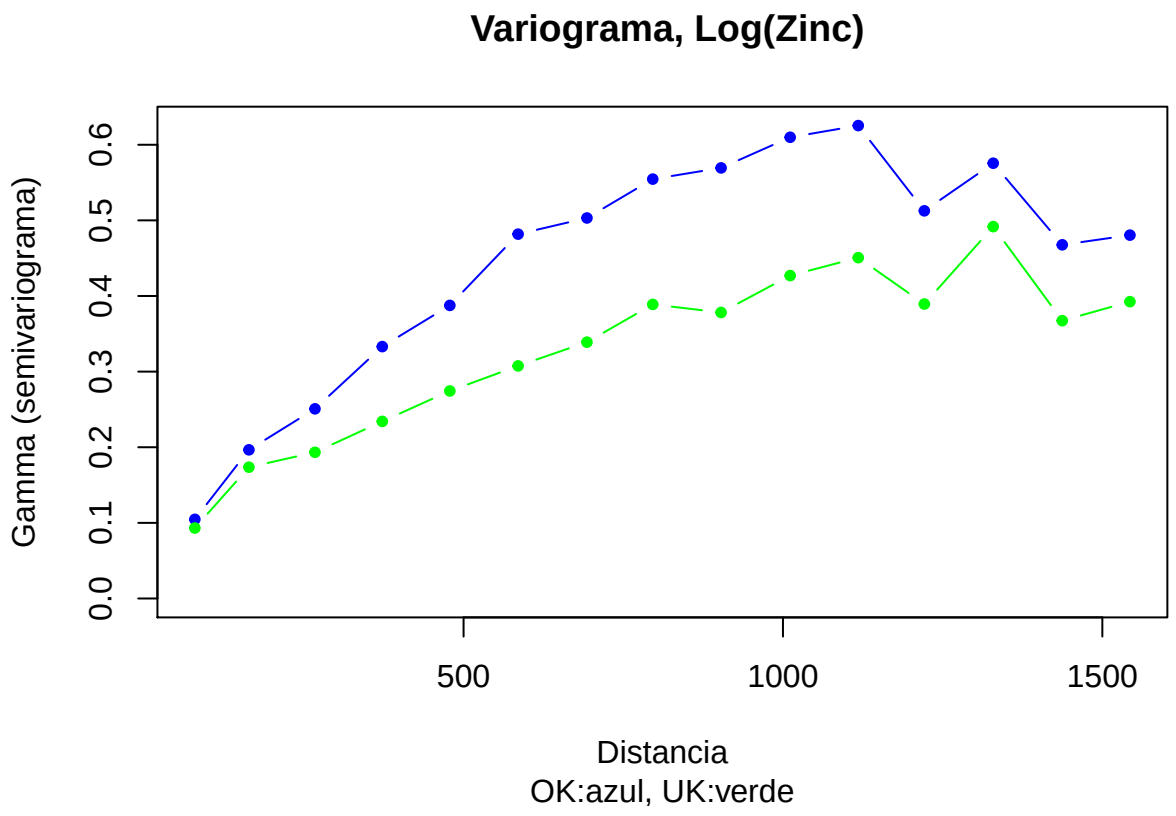
quito el efecto de las coordenadas En verde hago una estimación del universal, donde las gammas no son iguales

```
lead.res.variogram = variogram(log(lead)~x+y, meuse)

comparar.vgm <- data.frame(np = lead.variogram$np,
                           dist = lead.variogram$dist,
                           gamma.ok=lead.variogram$gamma,
                           gamma.uk=lead.res.variogram$gamma,
                           gamma.dif = lead.variogram$gamma - lead.res.variogram$gamma)

#Visualización de la comparación de los variogramas
comparar.vgm
```

```
##     np       dist  gamma.ok   gamma.uk  gamma.dif
## 1   57   79.29244 0.1046520 0.09315469 0.01149736
```

```
## 2  299  163.97367 0.1965929 0.17362913 0.02296381
## 3  419  267.36483 0.2507668 0.19329429 0.05747254
## 4  457  372.73542 0.3330690 0.23417580 0.09889318
## 5  547  478.47670 0.3875716 0.27441694 0.11315464
## 6  533  585.34058 0.4817750 0.30754417 0.17423082
## 7  574  693.14526 0.5031432 0.33894888 0.16419436
## 8  564  796.18365 0.5545787 0.38887646 0.16570219
## 9  589  903.14650 0.5693882 0.37821778 0.19117042
## 10 543 1011.29177 0.6098806 0.42698867 0.18289197
## 11 500 1117.86235 0.6253271 0.45068296 0.17464417
## 12 477 1221.32810 0.5126165 0.38935784 0.12325869
## 13 452 1329.16407 0.5755737 0.49172019 0.08385356
## 14 457 1437.25620 0.4676728 0.36738635 0.10028649
## 15 415 1543.20248 0.4804887 0.39250469 0.08798398
```

```
plot(comparar.vgm$gamma.ok ~ comparar.vgm$dist, pch=20, col="blue", type="b",
     xlab="Distancia", ylab="Gamma (semivariograma)",
     ylim=c(0,max(comparar.vgm$gamma.ok, comparar.vgm$gamma.uk)),
     main = " Variograma, Log(Zinc)", sub="OK:azul, UK:verde")

points(comparar.vgm$gamma.uk  ~ comparar.vgm$dist, pch=20, col="green", type="b")
```



Dadas las diferencias, mantenemos el variograma asociado a los residuos MCO.

Ajuste a un modelo teórico

Primero seleccionamos el mejor modelo en función de la suma de cuadrados del error.

```
attributes(fit.variogram(lead.res.variogram, model=vgm(1, "Sph", 900, 1)))$SSErr
```

```
## [1] 1.321783e-05
```

```
attributes(fit.variogram(lead.res.variogram, model=vgm(1, "Pen", 900, 1)))$SSErr
```

```
## [1] 1.300839e-05
```

```
attributes(fit.variogram(lead.res.variogram, model=vgm(1, "Gau", 900, 1)))$SSErr
```

```
## [1] 2.725643e-05
```

```
attributes(fit.variogram(lead.res.variogram, model=vgm(1, "Cir", 900, 1)))$SSErr
```

```
## [1] 1.38133e-05
```

```
attributes(fit.variogram(lead.res.variogram, model=vgm(1, "Exp", 900, 1)))$SSErr
```

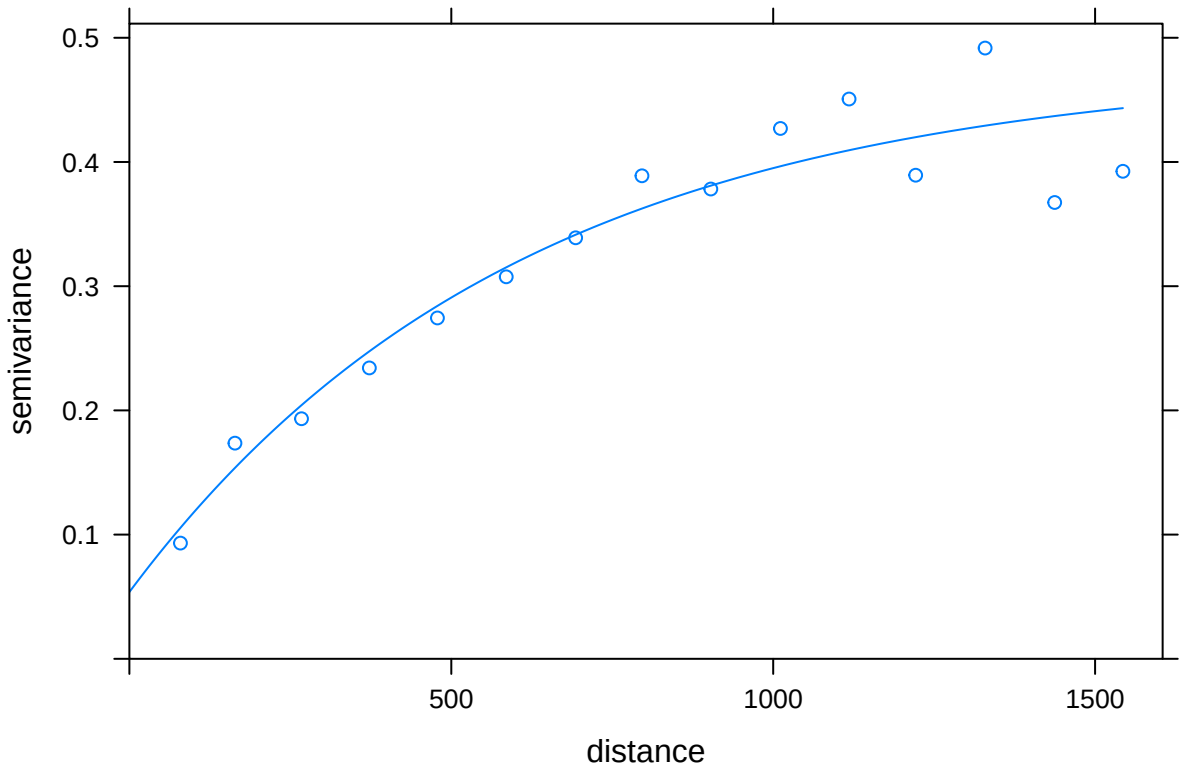
```
## [1] 1.201135e-05
```

Seleccionamos el modelo exponencial porque es el que tiene menor error. Lo almacenamos en un objeto "lead.res.fit"

```
lead.res.fit <- fit.variogram(lead.res.variogram, model = vgm(1, "Exp", 900, 1))
lead.res.fit
```

```
##   model     psill    range
## 1   Nug 0.0537787   0.0000
## 2   Exp 0.4230569 608.4942
```

```
plot(lead.res.variogram, lead.res.fit)
```



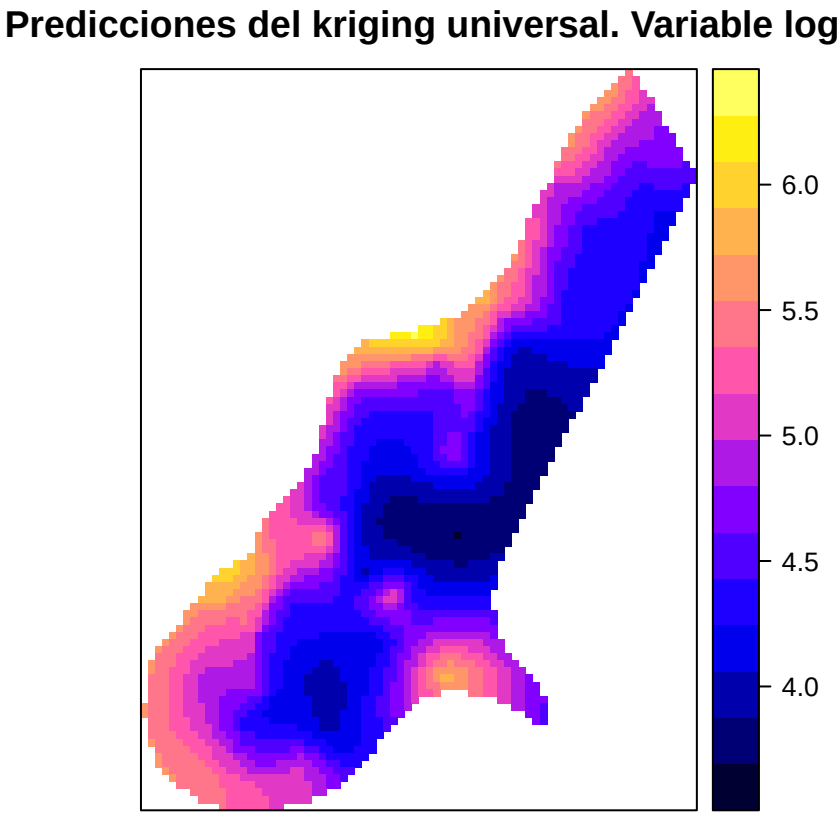
4.3 Predicciones Kriging Universal

Se crea el objeto kriging universal para log(lead), y se representa gráficamente las predicciones en el fichero rejilla.

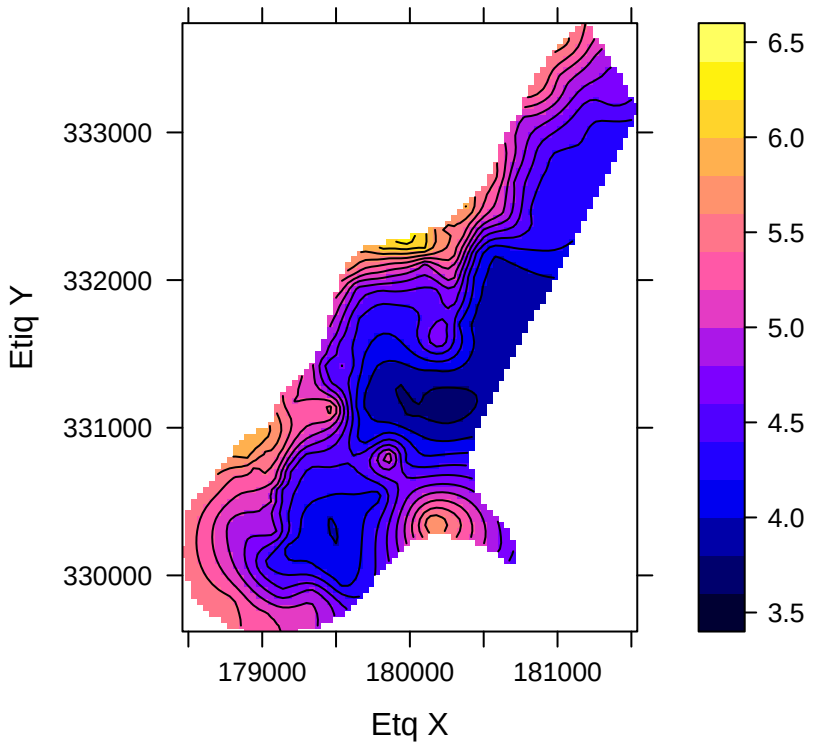
```
lead.ukriged = krige(log(lead)~x+y, meuse, meuse.grid, model = lead.res.fit)
```

```
## [using universal kriging]
```

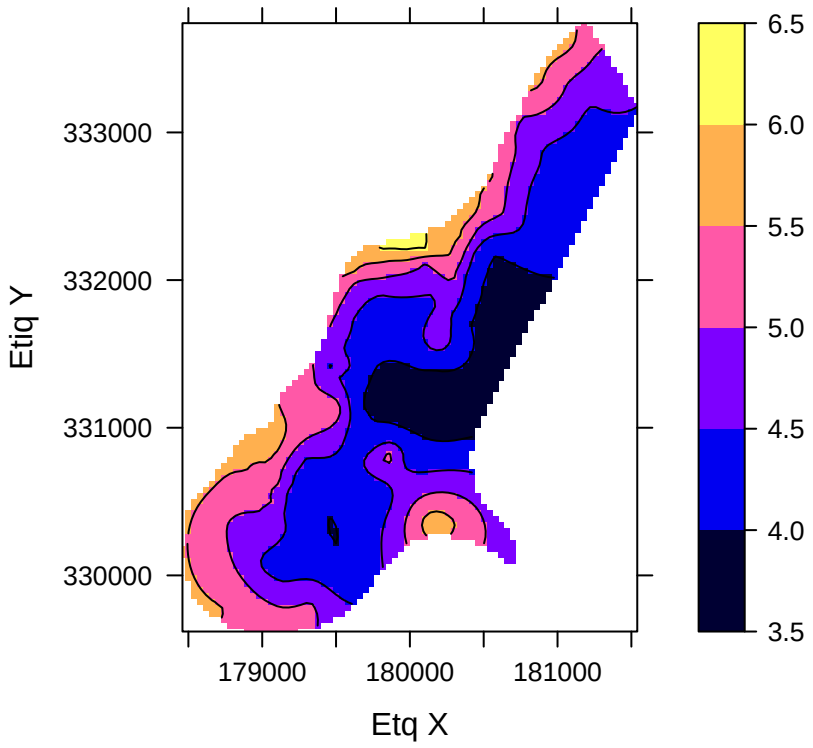
```
spplot(lead.ukriged["var1.pred"], main="Predicciones del kriging universal. Variable log(lead)")
```




```
spplot(lead.ukriged, zcol="var1.pred", pretty=T, contour=T, col.regions=bpy.colors(64),
       main="Predicciones del kriging universal. Variable log(lead)",
       xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))
```




```
spplot(lead.ukriged, zcol="var1.pred", pretty=T, contour=T, col.regions=bpy.colors(64),
       main="Predicciones del kriging universal. Variable log(lead)",
       xlab="Etq X",ylab="Etiq Y", scales=list(draw=T), cuts=8)
```



4.4 Cálculo y representación de diferencias entre kriging ordinario y universal

Creamos un data.frame para evaluar las diferencias

```
dif.uk.ok <- data.frame(dif.pred = lead.ukriged$var1.pred - lead.kriged$var1.pred,
                        dif.var = lead.ukriged$var1.var - lead.kriged$var1.var)
summary(dif.uk.ok)
```

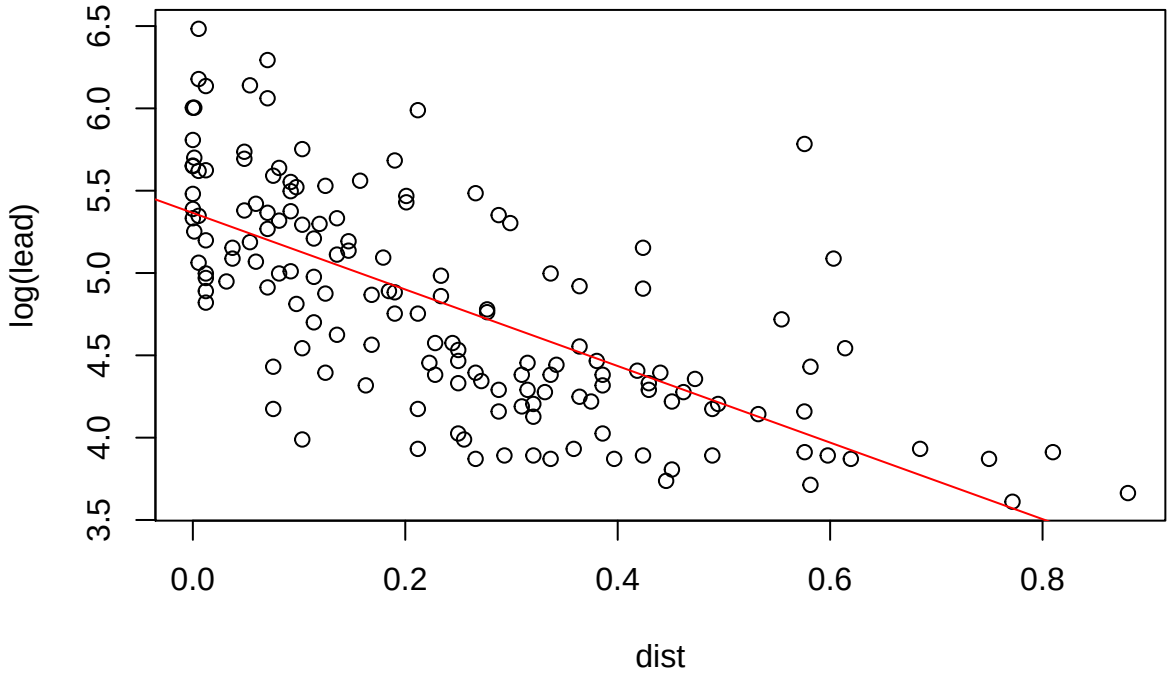
```
##     dif.pred           dif.var
##  Min.   :-0.36639   Min.   :-0.089812
##  1st Qu.:-0.02705   1st Qu.:-0.016143
##  Median :-0.00222   Median :-0.008992
##  Mean   :-0.01364   Mean   :-0.013835
##  3rd Qu.: 0.01689   3rd Qu.:-0.005640
##  Max.   : 0.44091   Max.   : 0.001461
```

Se observa que las predicciones son similares, aunque las varianzas de las estimaciones (y por tanto, los errores de estimación) son menores a través del kriging universal.

5 Kriging deriva externa para la variable objetivo con predictor distancia al rio

5.1 Búsqueda de la tendencia lineal en función de la distancia al río

```
plot(log(lead)~ dist, meuse)
abline(lm(formula=log(lead) ~ dist, data=meuse), col="red")
```




```
summary(lm(formula=log(lead) ~ dist, data=meuse))
```

```
##
## Call:
## lm(formula = log(lead) ~ dist, data = meuse)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13654 -0.33783 -0.04031  0.28741  1.75759
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.36505    0.06129   87.54   <2e-16 ***
## dist        -2.32483    0.19735  -11.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4842 on 153 degrees of freedom
## Multiple R-squared:  0.4756, Adjusted R-squared:  0.4722
## F-statistic: 138.8 on 1 and 153 DF,  p-value: < 2.2e-16
```

```
summary(lm(formula=log(lead) ~ coordinates(meuse)+dist, data=meuse))
```

```
##
## Call:
## lm(formula = log(lead) ~ coordinates(meuse) + dist, data = meuse)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09101 -0.30693 -0.04578  0.30847  1.76130
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.678e+00  1.311e+01  -0.204   0.8384
## coordinates(meuse)x -3.018e-04  1.196e-04  -2.524   0.0126 *
## coordinates(meuse)y  1.879e-04  8.612e-05   2.181   0.0307 *
## dist                  -2.007e+00  2.365e-01  -8.484 1.86e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4773 on 151 degrees of freedom
## Multiple R-squared:  0.497,  Adjusted R-squared:  0.487
## F-statistic: 49.73 on 3 and 151 DF,  p-value: < 2.2e-16
```

Parece más adecuado el primero, dado que la capacidad de explicación de las coordenadas es muy débil.

5.2 Construcción del variograma (de los residuos) muestral y ajuste modélo teórico

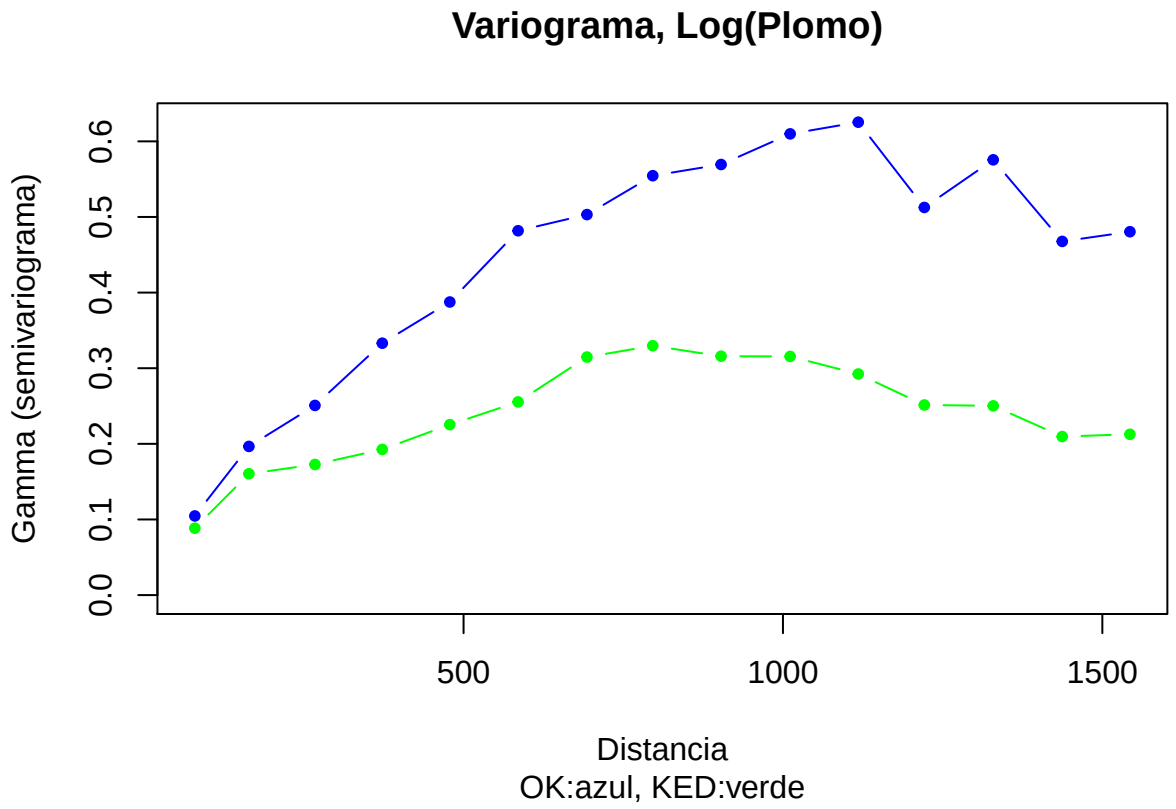
Se crea el variograma experimental o muestral de los residuos (objeto "lead.resdist.vgm")" y se compara con el semivariograma muestral obtenido para los datos originales, almacenado en el objeto "lead.variogram"

```
lead.resdist.vgm = variogram(log(lead)~dist, meuse)

rm(comparar.vgm) # Para eliminar el objeto ya existente con este nombre

comparar.vgm <- data.frame(np = lead.variogram$np,
                           dist = lead.variogram$dist,
                           gamma.ok = lead.variogram$gamma,
                           gamma.ked = lead.resdist.vgm$gamma,
                           gamma.dif = lead.variogram$gamma - lead.resdist.vgm$gamma)

#Visualización de la comparación de los variogramas
comparar.vgm
```

```
##     np       dist  gamma.ok gamma.ked  gamma.dif
## 1   57   79.29244 0.1046520 0.0884687 0.01618335
## 2  299  163.97367 0.1965929 0.1604258 0.03616715
## 3  419  267.36483 0.2507668 0.1727089 0.07805790
## 4  457  372.73542 0.3330690 0.1926428 0.14042614
## 5  547  478.47670 0.3875716 0.2253824 0.16218915
## 6  533  585.34058 0.4817750 0.2554218 0.22635322
```

```
## 7  574  693.14526 0.5031432 0.3147706 0.18837269
## 8  564  796.18365 0.5545787 0.3297327 0.22484594
## 9  589  903.14650 0.5693882 0.3158589 0.25352932
## 10 543 1011.29177 0.6098806 0.3155785 0.29430209
## 11 500 1117.86235 0.6253271 0.2923380 0.33298908
## 12 477 1221.32810 0.5126165 0.2513888 0.26122774
## 13 452 1329.16407 0.5755737 0.2502521 0.32532164
## 14 457 1437.25620 0.4676728 0.2096312 0.25804164
## 15 415 1543.20248 0.4804887 0.2126189 0.26786977
```

```
plot(comparar.vgm$gamma.ok ~ comparar.vgm$dist, pch=20, col="blue", type="b",
     xlab="Distancia", ylab="Gamma (semivariograma)",
     ylim=c(0,max(comparar.vgm$gamma.ok, comparar.vgm$gamma.ked)),
     main = " Variograma, Log(Plomo)", sub="OK:azul, KED:verde")

points(comparar.vgm$gamma.ked  ~ comparar.vgm$dist, pch=20, col="green", type="b")
```



Dadas las diferencias, mantenemos el variograma asociado a los residuos MCO.

5.3 Ajuste a un modelo teórico

Primero seleccionamos el mejor modelo en función de la suma de cuadrados del error.

```
attributes(fit.variogram(lead.resdist.vgm, model=vgm(1, "Sph", 900, 1)))$SSErr
```

```
## [1] 1.691364e-05
```

```
attributes(fit.variogram(lead.resdist.vgm, model=vgm(1, "Pen", 900, 1)))$SSErr
```

```
## [1] 1.712368e-05
```

```
attributes(fit.variogram(lead.resdist.vgm, model=vgm(1, "Gau", 900, 1)))$SSErr
```

```
## [1] 2.644922e-05
```

```
attributes(fit.variogram(lead.resdist.vgm, model=vgm(1, "Cir", 900, 1)))$SSErr
```

```
## [1] 1.688507e-05
```

```
attributes(fit.variogram(lead.resdist.vgm, model=vgm(1, "Exp", 900, 1)))$SSErr
```

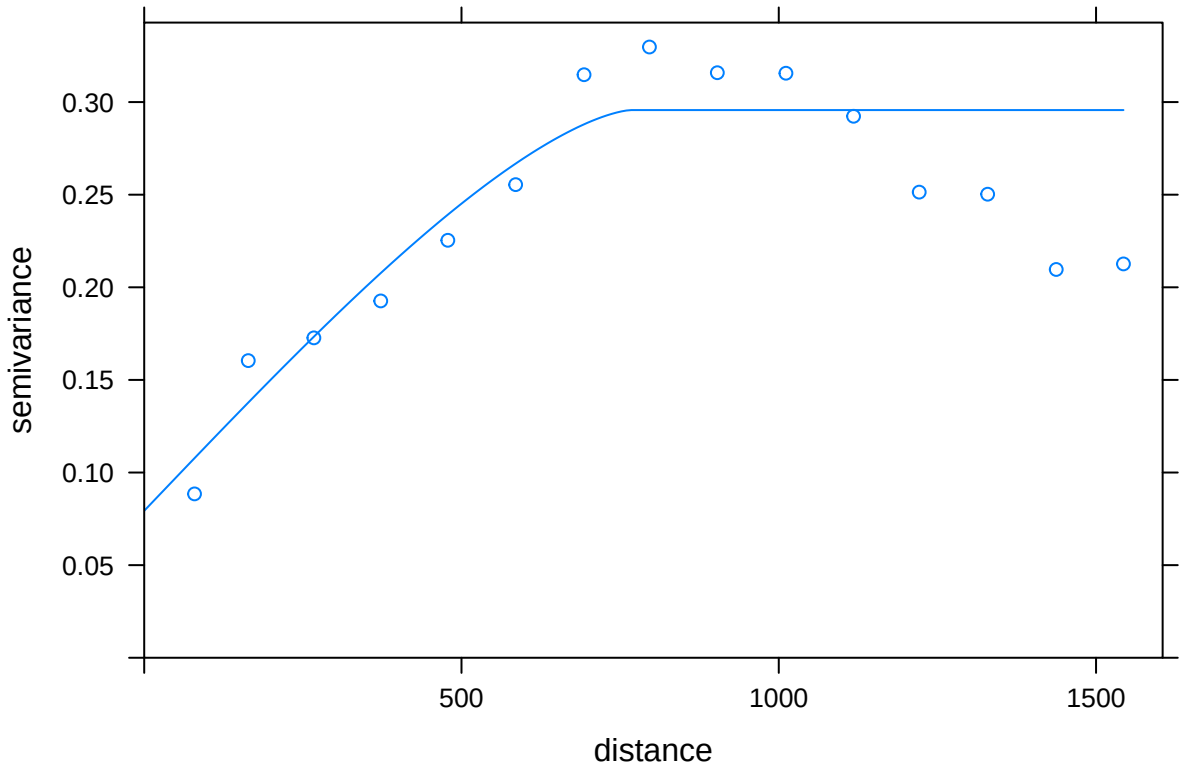
```
## [1] 1.757256e-05
```

Seleccionamos el modelo Circular y lo almacenamos en un objeto "lead.resdist.fit"

```
lead.resdist.fit <- fit.variogram(lead.resdist.vgm, model = vgm(1, "Cir", 900, 1))

lead.resdist.fit
```

```
##   model      psill    range
## 1   Nug 0.07934696   0.0000
## 2   Cir 0.21634195 767.3363
```

```
plot(lead.resdist.vgm, lead.resdist.fit)
```



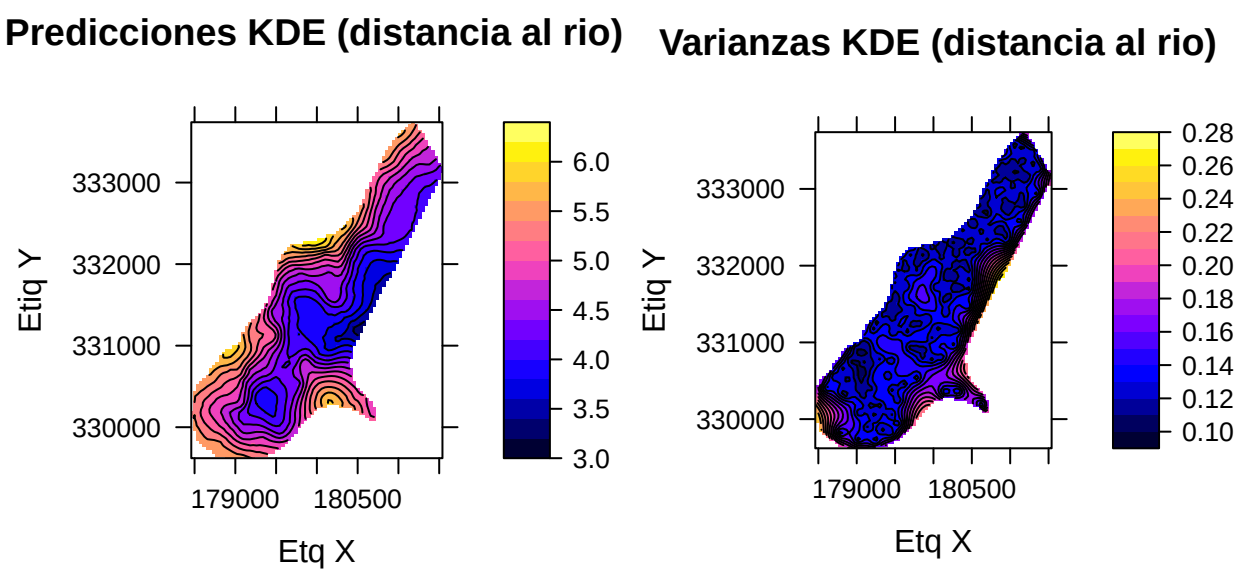
5.4 Predicciones Kriging con Deriva Externa

Se crea el objeto kriging con Deriva Externa para log(lead), y se representa gráficamente las predicciones en el fichero rejilla.

```
lead.dekriged = krige(log(lead)~ dist, meuse, meuse.grid, model = lead.resdist.fit)
```

```
## [using universal kriging]
```

```
plot.pred.dekriged <- spplot(lead.dekriged, zcol="var1.pred", pretty=T, contour=T,
                             col.regions=bpy.colors(64),
                             main="Predicciones KDE (distancia al rio)",
                             xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))

plot.var.dekriged <- spplot(lead.dekriged, zcol="var1.var", pretty=T, contour=T,
                            col.regions=bpy.colors(64),
                            main="Varianzas KDE (distancia al rio)",
                            xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))

print(plot.pred.dekriged, split=c(1,1,2,1), more =T)

print(plot.var.dekriged, split=c(2,1,2,1), more =F)
```



5.5 Cálculo y representación de diferencias entre kriging con deriva externa y universal

A continuación calcularemos la diferencia entre kriging universal y con deriva externa. No vamos a comparar con kriging ordinario porque en el apartado anterior ya vimos que kriging universal presentaba menor error.

Recorrido de las predicciones

```
range(lead.ukriged$var1.pred, lead.dekriged$var1.pred)
```

```
## [1] 3.288030 6.278931
```

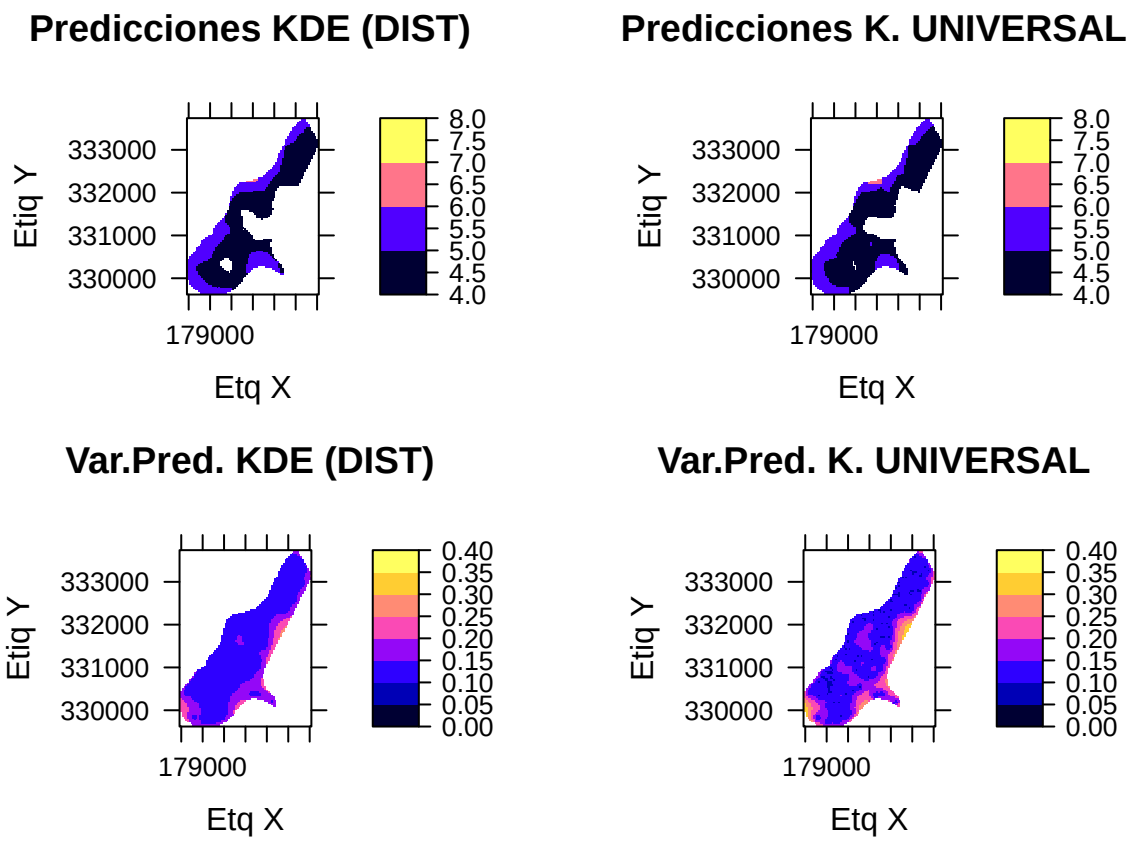
Recorrido de las varianzas de las estimaciones

```
range(lead.ukriged$var1.var, lead.dekriged$var1.var)
```

```
## [1] 0.08340877 0.33252016
```

Parámetros "at" para los gráficos comparativos

```
at.pred = 4:8
at.var = seq(0, 0.4, by=0.05)

plot.1 <-spplot(lead.dekriged , zcol="var1.pred", pretty=T, contour=F,
                col.regions=bpy.colors(64),
                main = "Predicciones KDE (DIST)",
                xlab="Etq X", ylab="Etiq Y", scales=list(draw=T), at=at.pred)

plot.2 <-spplot(lead.dekriged , zcol="var1.var", pretty=T, contour=F,
                col.regions=bpy.colors(64), main = "Var.Pred. KDE (DIST)",
                xlab="Etq X", ylab="Etiq Y", scales=list(draw=T), at=at.var)

plot.3 <- spplot(lead.ukriged, zcol="var1.pred", pretty=T, contour=F,
                 col.regions=bpy.colors(64), main="Predicciones K. UNIVERSAL",
                 xlab="Etq X", ylab="Etiq Y", scales=list(draw=T), at=at.pred)

plot.4 <- spplot(lead.ukriged, zcol="var1.var", pretty=T, contour=F,
                 col.regions=bpy.colors(64),
                 main="Var.Pred. K. UNIVERSAL",
                 xlab="Etq X", ylab="Etiq Y", scales=list(draw=T), at=at.var)

print(plot.1, split=c(1,1,2,2), more =T)

print(plot.2, split=c(1,2,2,2), more =T)

print(plot.3, split=c(2,1,2,2), more =T)

print(plot.4, split=c(2,2,2,2), more =F)
```



Diferencias entre las predicciones y los errores de estimación

```
# Diferencia en las predicciones
summary(lead.ukriged$var1.pred - lead.dekriged$var1.pred)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.5091718 -0.0542067  0.0056485  0.0003188  0.0705964  0.5471538
```

```
# Diferencia en las varianzas de las predicciones
summary(lead.ukriged$var1.var  - lead.dekriged$var1.var)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.029110 -0.004723  0.003568  0.008114  0.017909  0.086038
```

Se observa predicciones similares, con errores de estimación ligeramente mayores en el kriging universal.

6 Kriging residual directo para la variable objetivo con predictor distancia al rio

6.1 Paso 1

Estimación de los parámetros que determinan la deriva a través del método de mínimos cuadrados ordinarios (MCO)

Se aplica la función "lm" (linear model) indicando (var_objetivo~var_explicativas, conjunto_datos)

```
deriva=lm(log(lead)~dist, meuse)
summary(deriva)
```

```
##
## Call:
## lm(formula = log(lead) ~ dist, data = meuse)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13654 -0.33783 -0.04031  0.28741  1.75759
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.36505    0.06129   87.54   <2e-16 ***
## dist        -2.32483    0.19735  -11.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4842 on 153 degrees of freedom
## Multiple R-squared:  0.4756, Adjusted R-squared:  0.4722
## F-statistic: 138.8 on 1 and 153 DF,  p-value: < 2.2e-16
```

Se salvan los residuos ordinarios en una variable que se incluye en el conjunto de datos original (se hace copia para mantener el inicial)

```
R0lnplomo<-residuals(deriva)
meuse2=meuse
meuse2@data=cbind(meuse2@data, R0lnplomo)
```

6.2 Paso2

Kriging ordinario sobre los residuos del método de mínimos cuadrados ordinarios (MCO)

Se inicia el proceso, estimando el modelo de semivariograma teórico.

Con las órdenes:

```
R0.vgm = variogram(R0lnplomo~1,meuse2)
attributes(fit.variogram(R0.vgm, model=vgm(0.15, "Sph", 900, 0.1)))$SSErr
attributes(fit.variogram(R0.vgm, model=vgm(0.15, "Pen", 900, 0.1)))$SSErr
attributes(fit.variogram(R0.vgm, model=vgm(0.15, "Gau", 900, 0.1)))$SSErr
attributes(fit.variogram(R0.vgm, model=vgm(0.15, "Cir", 900, 0.1)))$SSErr
attributes(fit.variogram(R0.vgm, model=vgm(0.15, "Exp", 900, 0.1)))$SSErr
```

Se podría ajustar el modelo que coincidirá con el obtenido en el método anterior: *lead.resdist.fit*

Una vez obtenido el variograma teórico de los residuos, se realiza el krigeado de los residuos, considerando el conjunto de datos grid o rejilla inicial.

```
R0.kriged = krige(R0lnplomo~1, meuse2, meuse.grid, model = lead.resdist.fit)
```

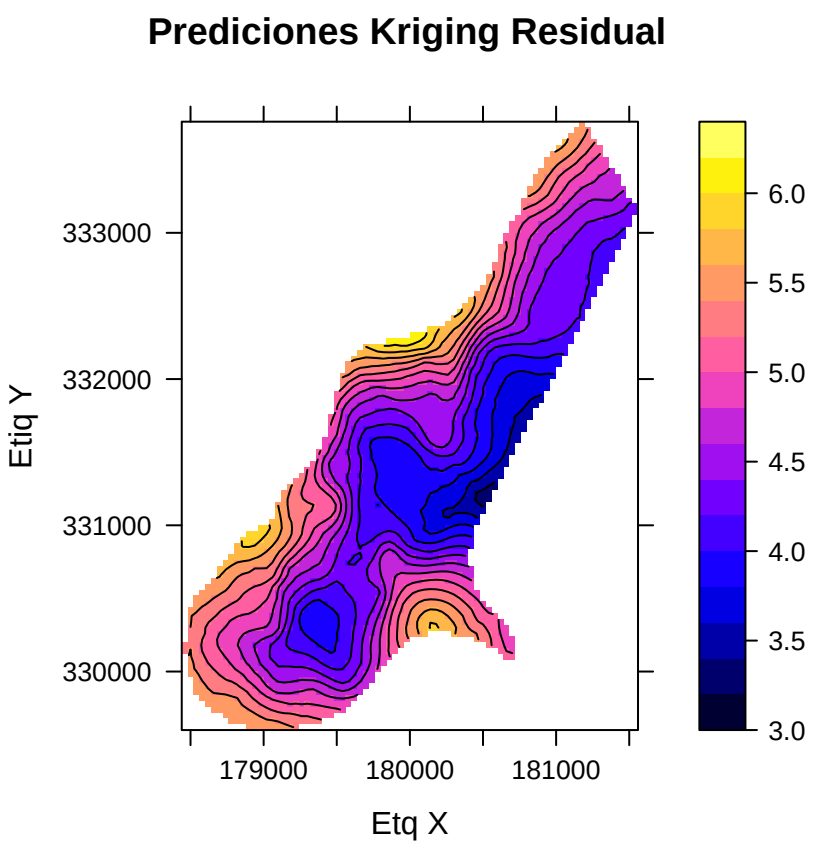
```
## [using ordinary kriging]
```

6.3 Paso 3

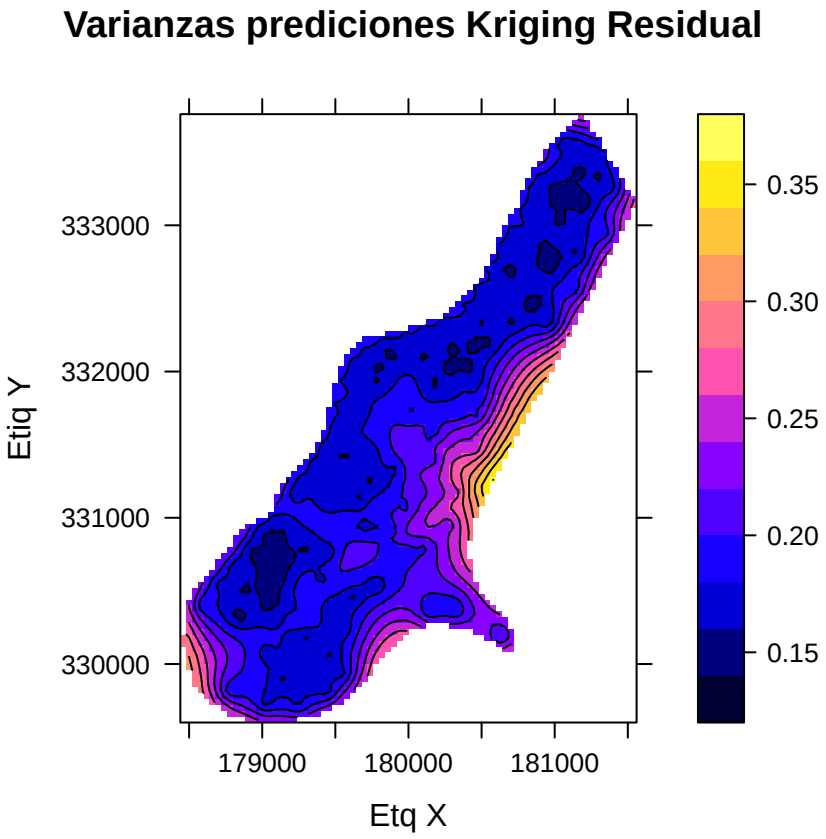
Cálculo de la predicción del proceso como suma de la deriva ajustada y el ajuste de los residuos realizados a través del Kriging

La predicción se realiza sobre la rejilla, para lo cual se hace previamente una copia y se incrustan dichas predicciones

```
Predderiva = predict(deriva, meuse.grid)    # Predicción de la deriva
Predresid = R0.kriged@data$var1.pred        # Predicción kriging de los residuos
Predfinal = Predderiva+Predresid            # Predicción conjunta

var.predderiva = predict(deriva, meuse.grid, se.fit=TRUE)$se.fit
var.predresid  = R0.kriged@data$var1.var
var.predfinal  = var.predderiva + var.predresid

meuse2.grid=meuse.grid
meuse2.grid@data=cbind(meuse2.grid@data, Predfinal, var.predfinal)

spplot(meuse2.grid, zcol="Predfinal", pretty=T, contour=T, col.regions=bpy.colors(64),
   main="Prediciones Kriging Residual", xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))
```




```
spplot(meuse2.grid, zcol="var.predfinal", pretty=T, contour=T, col.regions=bpy.colors(64),
   main="Varianzas prediciones Kriging Residual", xlab="Etq X", ylab="Etiq Y", scales=list(draw=T))
```



6.4 Cálculo y representación de diferencias entre los dos métodos kriging con deriva externa

Determinamos el recorrido de las predicciones y las varianzas de las estimaciones

```
range(meuse2.grid$Predfinal, lead.dekriged$var1.pred)
```

```
## [1] 3.286638 6.140119
```

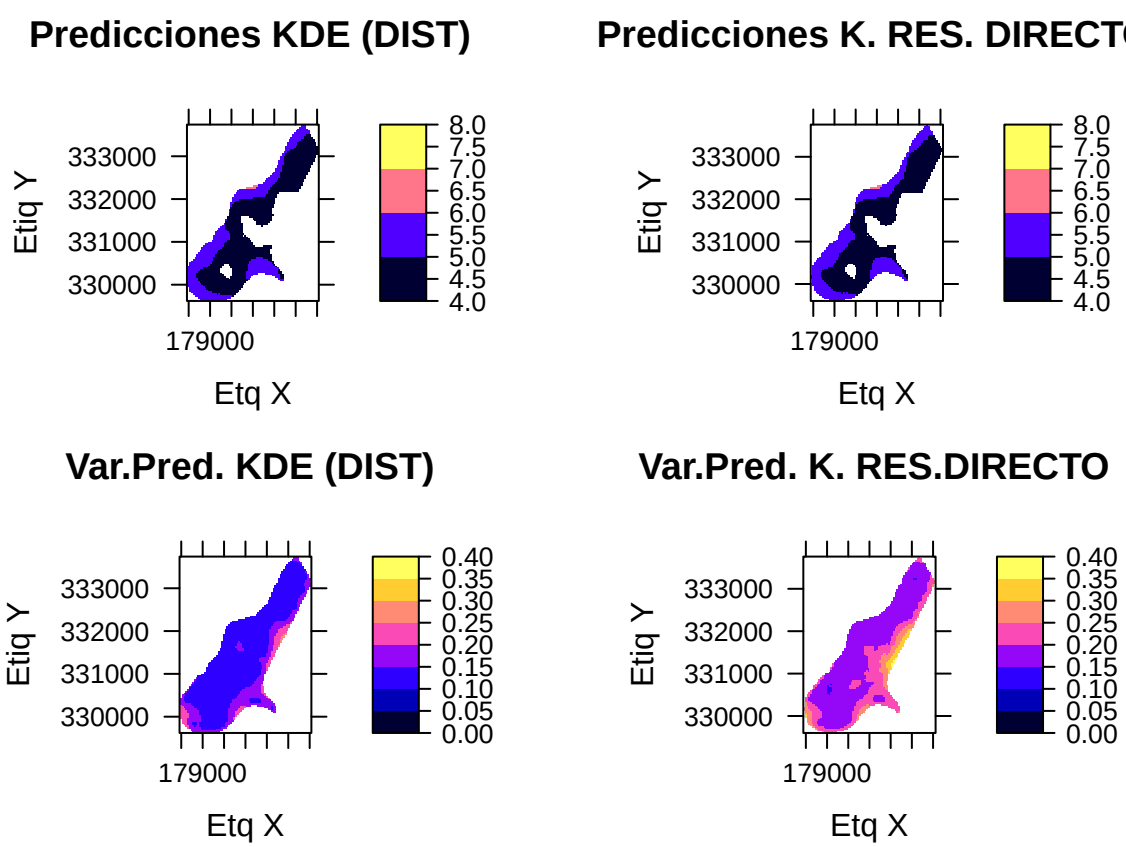
```
range(meuse2.grid$var.predfinal, lead.dekriged$var1.var)
```

```
## [1] 0.1042200 0.3601859
```

Determinamos los parámetros "at" para los gráficos comparativos

```
at.pred = 4:8
at.var = seq(0, 0.4, by=0.05)

plot.1 <-spplot(lead.dekriged , zcol="var1.pred", pretty=T, contour=F, col.regions=bpy.colors(64),
                main = "Predicciones KDE (DIST)", xlab="Etq X", ylab="Etiq Y",
                scales=list(draw=T), at=at.pred)

plot.2 <-spplot(lead.dekriged , zcol="var1.var", pretty=T, contour=F, col.regions=bpy.colors(64),
                main = "Var.Pred. KDE (DIST)", xlab="Etq X", ylab="Etiq Y",
                scales=list(draw=T), at=at.var)

plot.3 <- spplot(meuse2.grid, zcol="Predfinal", pretty=T, contour=F, col.regions=bpy.colors(64),
```

```
                  main="Predicciones K. RES. DIRECTO", xlab="Etq X", ylab="Etiq Y",
                  scales=list(draw=T), at=at.pred)

plot.4 <- spplot(meuse2.grid, zcol="var.predfinal", pretty=T, contour=F, col.regions=bpy.colors(64),
                  main="Var.Pred. K. RES.DIRECTO", xlab="Etq X", ylab="Etiq Y",
                  scales=list(draw=T), at=at.var)

print(plot.1, split=c(1,1,2,2), more =T)

print(plot.2, split=c(1,2,2,2), more =T)

print(plot.3, split=c(2,1,2,2), more =T)

print(plot.4, split=c(2,2,2,2), more =F)
```



Diferencias entre las predicciones

```
summary(meuse2.grid$Predfinal - lead.dekriged$var1.pred)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.460e-03 -1.132e-04 -1.913e-05 -5.303e-05  4.868e-05  6.231e-04
```

Diferencias entre los errores de estimación

```
summary(meuse2.grid$var.predfinal  - lead.dekriged$var1.var )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03851 0.04173 0.04968 0.05489 0.05944 0.13594
```

7 Conclusión

Se observa predicciones similares, con errores de estimación mayores en el kriging residual directo, por tanto el modelo que mejor resultados presenta es el kriging con deriva externa.