



**UNIVERSIDAD
DE GRANADA**

Universidad de Granada

Escuela Internacional de Posgrado

Máster en Estadística Aplicada

Materia: Modelos de Espacios Temporales.

Alumno: Francisco Javier Márquez Rosales

Tema 3: Estimación e Interpolación/Predicción.

Ejercicios:

Noviembre, 2022

Buscar algún conjunto de datos espaciales (datos correspondientes a alguna variable con distribución espacial) y realizar con dichos datos un análisis geoestadístico, incluyendo:

- ▶ Representación del conjunto de coordenadas de las observaciones.
- ▶ Cálculo y representación del variograma empírico.
- ▶ Ajuste de algunos variogramas teóricos (al menos dos modelos) y comparación.
- ▶ Mapa de interpolación mediante kriging (se sugiere aplicar Kriging Ordinario).

Solución:

Para este ejercicio usaremos los datos proporcionados por la ciudad de Nueva York Inicialmente, sobre ubicación de las Escuelas y variables asociadas. Del conjunto original nos quedaremos con las variables: latitud, longitud y Score (resultado a examen SAT de matemática).

Usaremos en la solución el paquete geoR.

Preparación de los datos

```
library(geoR)

## Warning: package 'geoR' was built under R version 4.1.3
## -----
## Analysis of Geostatistical Data
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
## geoR version 1.9-2 (built on 2022-08-09) is now loaded
## -----
```

Hacemos la lectura de la data, previamente descargada.y examinamos las variables seleccionada.

En primer lugar llemos la data de trabajo

```
datany <- read.csv('NY_schools.csv', header = T)
str(datany)
datany2 <- datany[,c(5,29,30)]
str(datany2)

## 'data.frame': 479 obs. of 3 variables:
## $ SAT.Math.Avg..Score: int 404 423 402 401 433 557 574 418 604 40
## 0 ...
## $ lat : num 40.7 40.7 40.7 NA 40.7 ...
## $ long : num -74 -74 -74 NA -74 ...
```

Convertimos ahora la data al formato espacial

```
datanygeo <- as.geodata(datany2, coords.col = 2:3, data.col = 1, data.names = NULL)
```

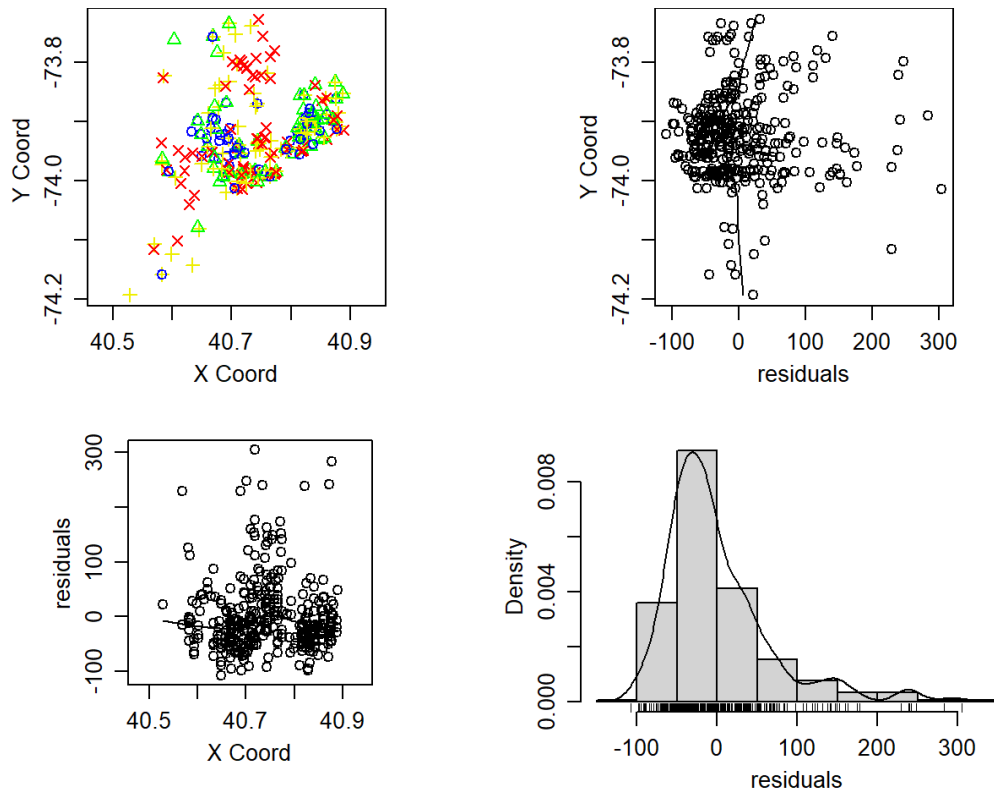
Hacemos un resumen de los datos obtenidos

```
summary(datanygeo)
## Number of data points: 340
##
## Coordinates summary
##          lat          long
## min 40.52823 -74.19215
## max 40.88837 -73.72691
##
## Distance summary
##          min          max
## 0.0000000 0.5132254
##
## Data summary
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 312.0000 376.7500 401.0000 419.9353 445.2500 735.0000
```

Representación del conjunto de coordenadas de las observaciones.

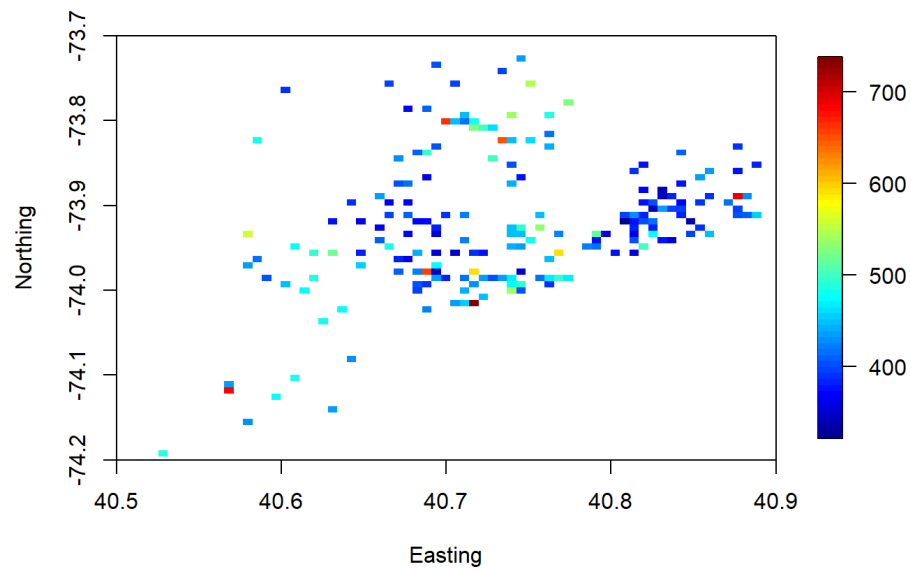
A continuación, generamos gráficos que nos describen las características principales de los datos.

```
plot(datanygeo, trend = "1st", lowess=TRUE)
```



Tambi[en, podemos genera un gráfico similar a una imagen donde las ubicaciones se discretizan en una cuadrícula, dado un vector de valores z de la variable resultado (score) asociados con ubicaciones bidimensionales los valores z se codifican como un nivel de color de una escala de color.

```
quilt.plot(datanygeo$coords, datanygeo$data, ny = ,
           ylim = c(-74.2, -73.7),
           xlim = c(40.5, 40.9), xlab = "Easting", ylab = "Northing")
```



Cálculo y representación del variograma empírico.

Obtenemos el variograma empírico a través de las siguientes comandos:

```
par(mfrow = c(1, 2), las = 1)

cloud1 <- variog(datanygeo, option = "cloud", max.dist=1)
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
bin1 <- variog(datanygeo, uvec=seq(0,1,l=11))
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
plot(cloud1, main = "Estimator clasico")

#plot(bin1, main = "classical estimator")
datanygeo.vario <- variog(datanygeo, max.dist = 1)
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
#plot(datanygeo.vario)
datanygeo.variot <- variog(datanygeo, trend = "1st", max.dist = 1)
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
plot(datanygeo.variot)
```

Y hacemos una representación gráfica (ver gráfico 1). Adicionalmente, con las siguientes sintaxis, los puntos de las nubes de variograma se pueden agrupar en clases de distancias ("contenedores") y mostrarse con un diagrama de caja para cada contenedor (ver gráfico 2).

```
bin1 <- variog(datanygeo,uvec = seq(0,1,l=11), bin.cloud = T)
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
par(mfrow = c(1,2))
plot(bin1, bin.cloud = T, main = "Estimator clasico")
```

Grafico 1.

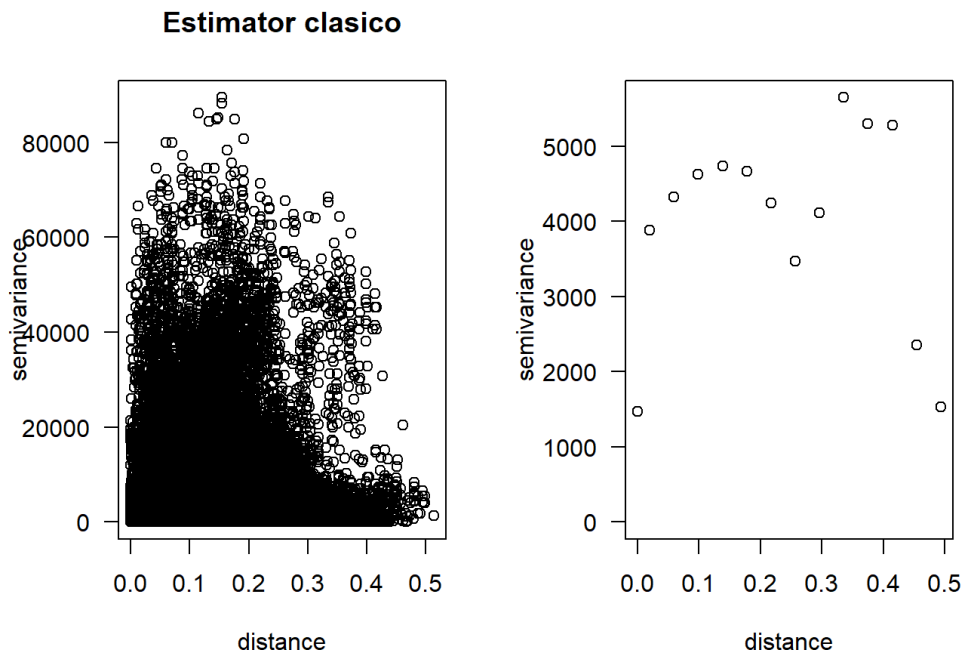
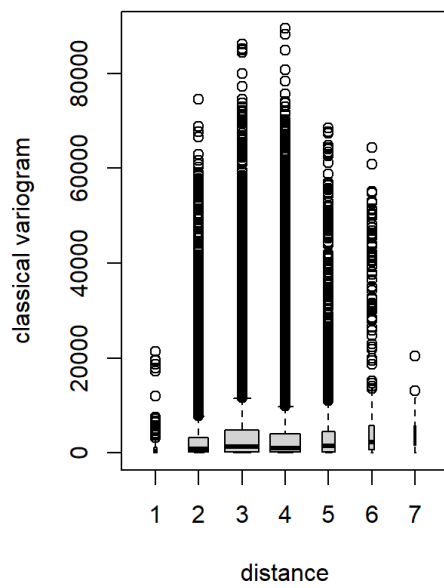
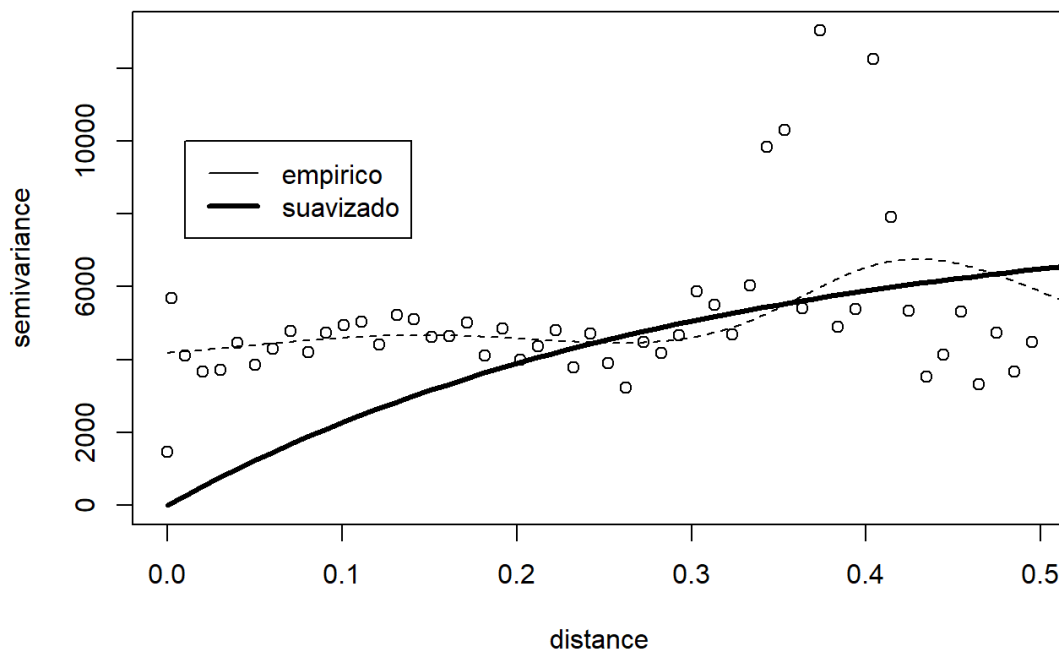


Grafico 2.



Los variogramas teóricos y empíricos se pueden trazar y comparar visualmente. Por ejemplo, la siguiente figura muestra el modelo de variograma teórico utilizado para simular los datos 'datany' y el variograma estimado suavizado.

```
bin1 <- variog(datanygeo, uvec = seq(0,1,l=100))
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
plot(bin1)
lines.variomodel(cov.model = "exp", cov.pars = c(8000,0.3), nugget = 0
, max.dist = 1, lwd = 3)
smooth <- variog(datanygeo, option = "smooth", max.dist = 1, n.points
= 400, kernel = "normal", band = 0.15)
## variog: computing omnidirectional variogram
## variog: co-located data found, adding one bin at the origin
lines(smooth, type = "l", lty = 2)
legend(0.01, 10000, c("empirico", "suavizado"), lty = c(1,1,2), lwd =
c(1,3,1))
```



Ajuste de algunos variogramas teóricos (al menos dos modelos) y comparación.

Obtenemos ahora los modelos ajustados.

```
datanygeo.vfit.exp <- variofit(datanygeo.vario)

## variofit: covariance model used is matern
## variofit: weights used: npairs
## variofit: minimisation function used: optim
## Warning in variofit(datanygeo.vario): initial values not provided -
  running the
## default search
## variofit: searching for best initial value ... selected values:
##           sigmasq  phi  tausq  kappa
## initial.value "3877.71" "0"  "775.54" "0.5"
## status        "est"    "est" "est"   "fix"
## loss value: 17869646153.811

datanygeo.vfit.mat1.5 <- variofit(datanygeo.vario, kappa = 1.5)

## variofit: covariance model used is matern
## variofit: weights used: npairs
## variofit: minimisation function used: optim
## Warning in variofit(datanygeo.vario, kappa = 1.5): initial values n
ot provided -
## running the default search
## variofit: searching for best initial value ... selected values:
##           sigmasq  phi  tausq  kappa
## initial.value "3877.71" "0"  "775.54" "1.5"
## status        "est"    "est" "est"   "fix"
## loss value: 17869646153.811

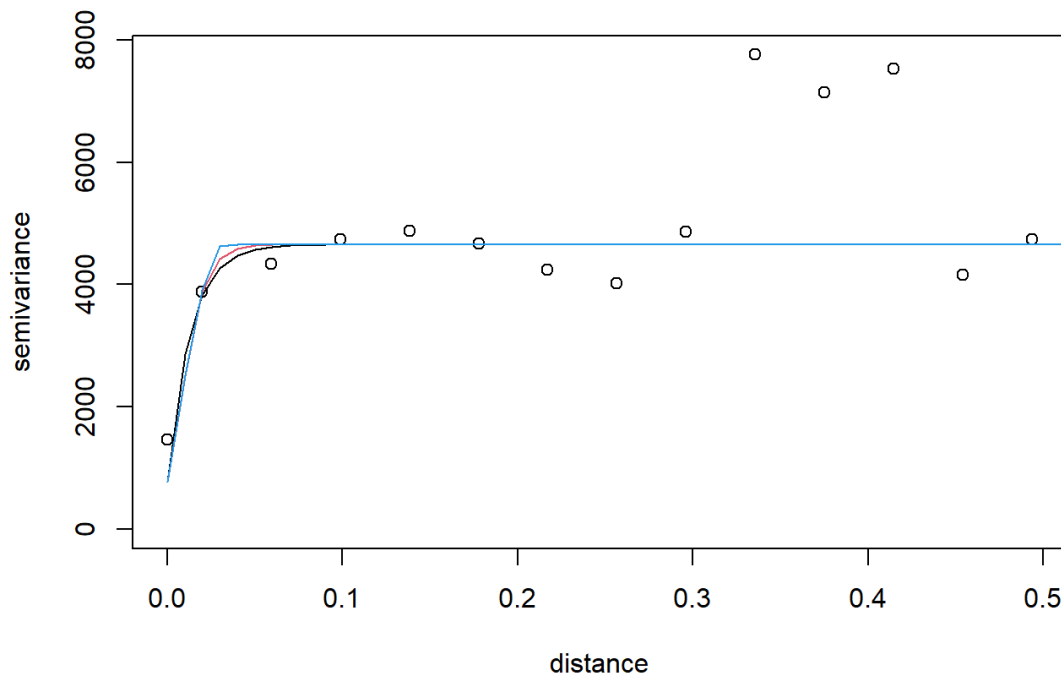
datanygeo.vfit.sph <- variofit(datanygeo.vario, cov.model = "sph")

## variofit: covariance model used is spherical
## variofit: weights used: npairs
## variofit: minimisation function used: optim
## Warning in variofit(datanygeo.vario, cov.model = "sph"): initial va
lues not
## provided - running the default search
## variofit: searching for best initial value ... selected values:
```

```
##          sigmasq  phi  tausq  kappa
## initial.value "3877.71" "0"  "775.54" "0.5"
## status        "est"    "est" "est"    "fix"
## loss value: 17869646153.811
```

Graficamos el resultado del ajuste de estos modelos

```
#par(mfrow = c(1, 2))
plot(datanygeo.vario)
lines(datanygeo.vfit.exp); lines(datanygeo.vfit.mat1.5, col = 2); line
s(datanygeo.vfit.sph, col = 4)
```



Mapa de interpolación mediante kriging (se sugiere aplicar Kriging Ordinario).

consideremos la predicción en cuatro ubicaciones etiquetadas con 1, 2, 3, 4 e indicadas a continuación.

```
loci <- matrix(c(40, 40.1, 40.2, 40.3, 40.7, 40.8, 40.9, 41), ncol=2)

kc4 <- krige.conv(datanygeo, locations = loci, krige = krige.control(o
bj.m = datanygeo.vfit.exp))
```

"Valores predichos"

```
kc4$predict
## [1] 437.0036 437.0036 437.0036 437.0036
```

"Varianza kriging"

```
kc4$krige.var
## [1] 4724.724 4724.724 4724.724 4724.724
```