

Práctica 1

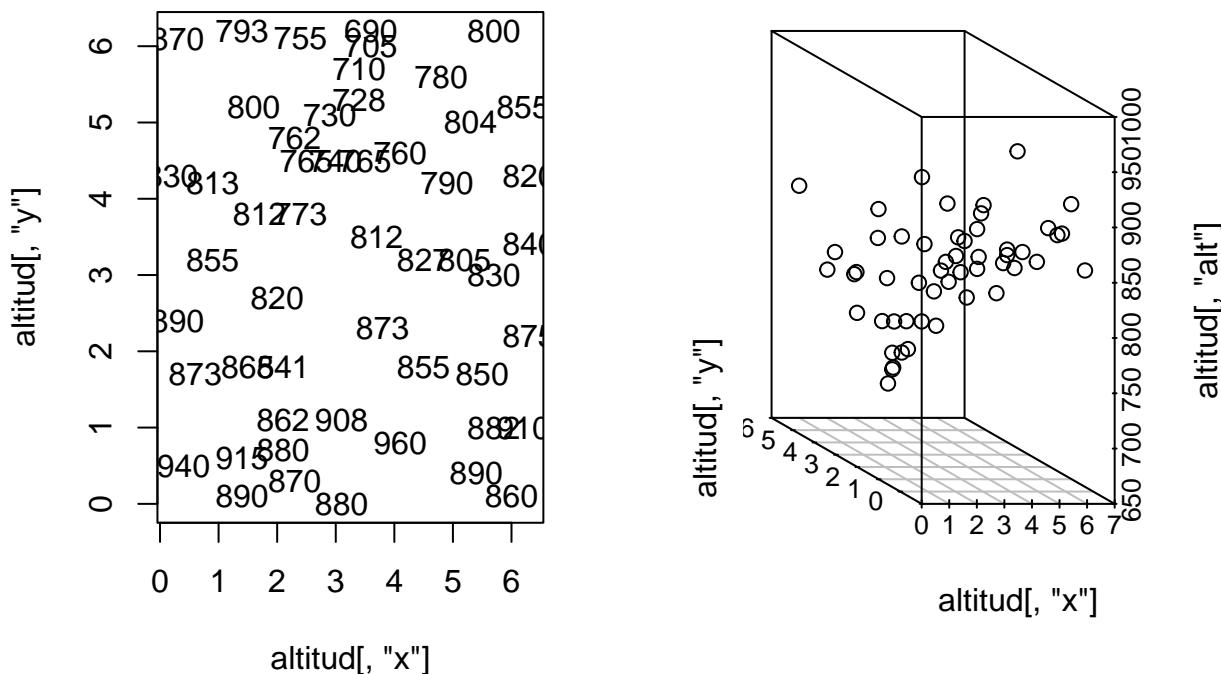
Juan Cantero Jimenez

4/4/2022

1. Lectura de los datos y análisis preliminar.

En el presente trabajo, se analizará un dataset que consta de 52 observaciones que recogen la altitud de una parcela de terreno. Los datos pueden ser introducidos en R, y explorados mediante los siguientes comandos.

```
source("altitud.r")
library(scatterplot3d)
par(mfrow = c(1,2))
plot(altitud[, "x"], altitud[, "y"], type="n")
text(altitud[, "x"], altitud[, "y"], altitud[, "alt"])
scatterplot3d(altitud[, "x"], altitud[, "y"], altitud[, "alt"], angle=160)
```



Si se observan los datos detenidamente, se podrá apreciar una ligera disminución en la altitud del terreno a lo largo del eje Y. Este comportamiento ocurre también en el eje X donde parece existir una depresión en el terreno respecto a la posición central de esta dimensión. Este comportamiento puede observarse de forma más directa si se representan las coordenadas, por separado, frente al valor de la variable en ese punto. Este tipo de gráficas puede ser obtenido gracias a la función plot.geodata del paquete geoR:

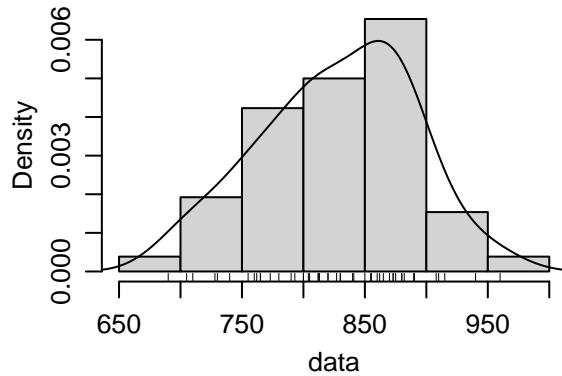
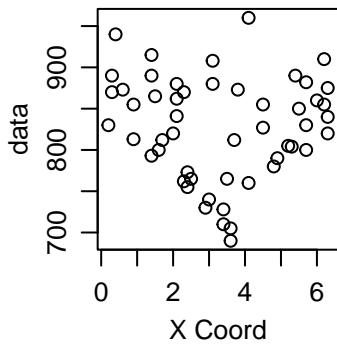
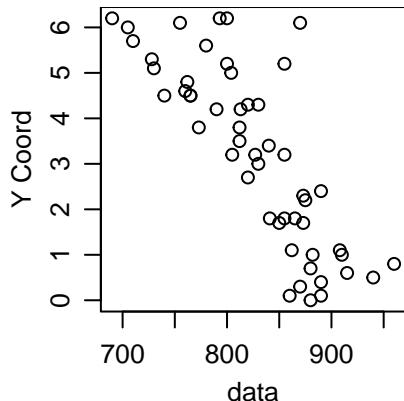
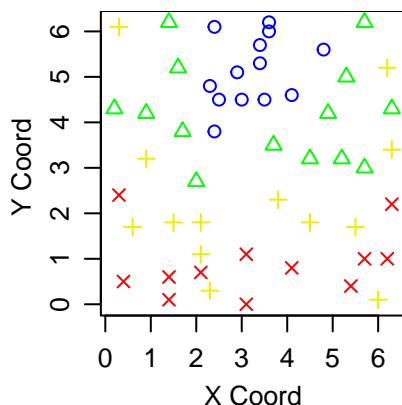
```
library(geoR)

## -----
## Analysis of Geostatistical Data
```

```

##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.8-1 (built on 2020-02-08) is now loaded
## -----
alt.geo<-as.geodata(altitud)
plot.geodata(alt.geo)

```



Si se presta atención al panel anterior, se podrá vislumbrar de forma más clara los comportamientos descritos anteriormente. En el gráfico superior derecha se puede apreciar una reducción sistemática en la altitud del terreno conforme se avanza en la coordenada y. Además esta tendencia se vuelve a repetir para los valores centrales de la coordenada X, aunque con la diferencia de que valores de altitud muy dispares coexisten para los mismo valores en la coordenada X. También es necesario prestar atención al gráfico superior izquierdo, que representa la superficie del terreno con las altitudes agrupadas, en el se puede apreciar una marcada depresión en el terreno con un mínimo aproximadamente en las coordenadas ($x = 3$ U, $y = 5$ U), en el que en forma de anillos concéntricos va aumentando la altitud. Estos resultados sugieren de la presencia de tendencia en los datos. Por último, se hace necesario hablar del gráfico inferior derecho, que muestra un histograma de la variable altitud, en el que se puede sospechar de una distribución normal para esta variable.

2. Análisis de la tendencia.

La determinación de superficies de tendencia en unos datos geoespaciales se puede obtener mediante la función `surf.ls()` del paquete spatial y puede evaluarse en un grid sobre una región con la función `trmat()`. La función `surf.ls()` requiere introducir el grado del polinomio que servirá como modelo matemático para la superficie de tendencia.

```

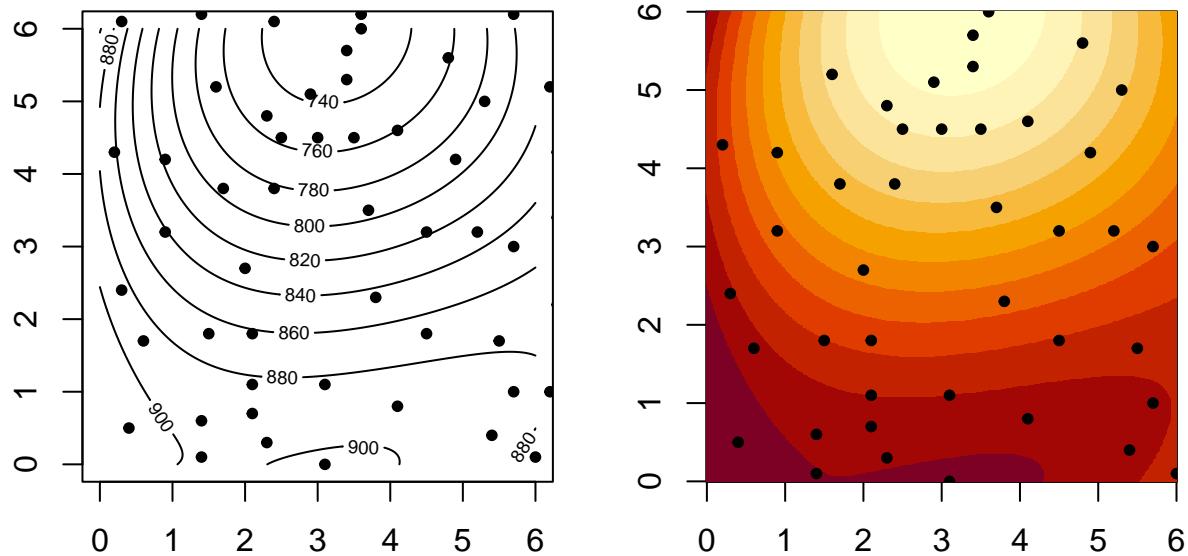
library(spatial)
alt.ls<-surf.ls(3,altitud[, "x"],altitud[, "y"],altitud[, "alt"])

```

```
alt.trsurf<-trmat(alt.ls, 0, 6, 0, 6, 600)
```

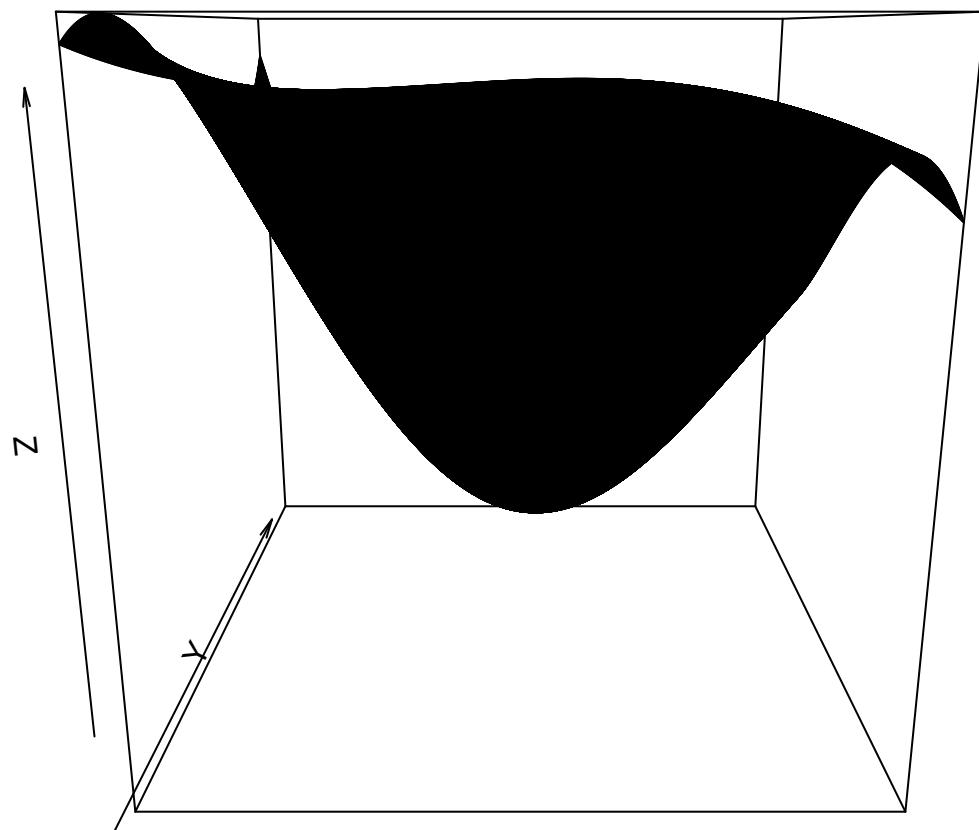
Esta superficie de tendencia puede visualizarse mediante gráficas de dos dimensiones

```
par(pty="s",mar=c(2,2,2,2), mfrow=c(1,2))
contour(alt.trsurf)
points(altitud[, "x"], altitud[, "y"], pch=20)
image(alt.trsurf)
points(altitud[, "x"], altitud[, "y"], pch=20)
```

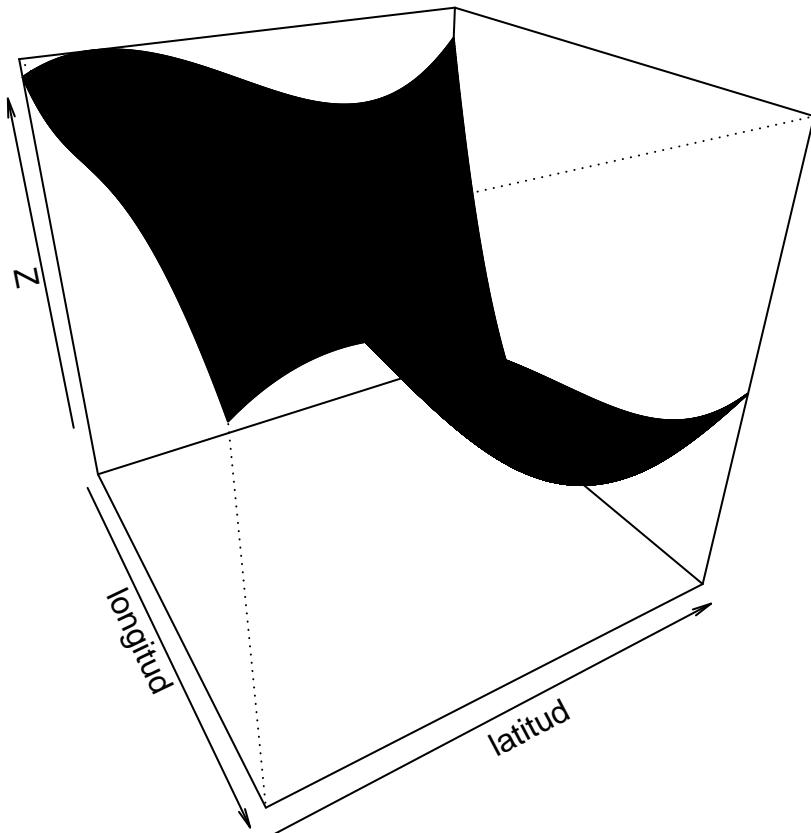


así como por gráficos que intentan dar una sensación de 3D a los datos

```
par(mar=c(0,0,0,0))  
persp(alt.trsurf)
```

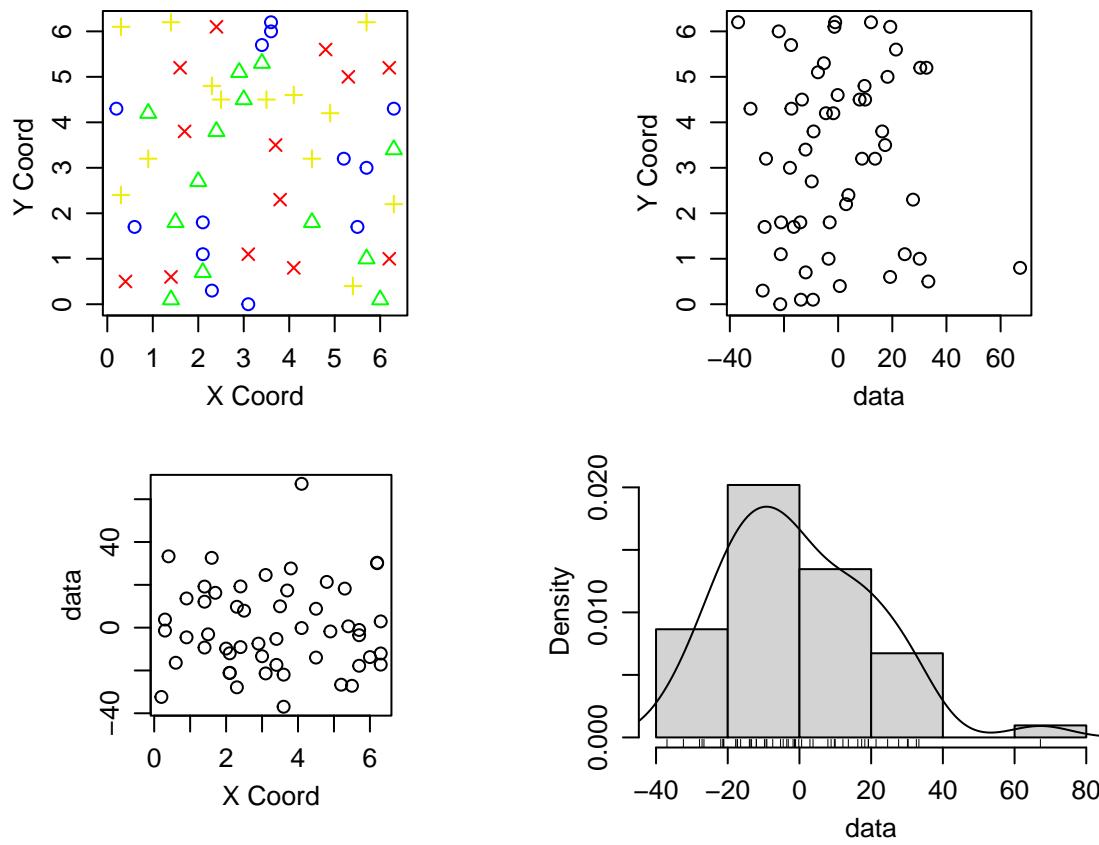


```
persp(alt.trsurf,theta=60,phi=30,col=2,ltheta=-20,shade=0.25,xlab="longitud",ylab="latitud")
```



En este caso particular, una superficie generada con un polinomio de grado 3 parece explicar de forma satisfactoria la tendencia que se aprecia en los datos. Eso se puede confirmar si observamos de nuevo los datos pero eliminando primero la tendencia.

```
alt.sin<-altitud[,3]-predict(alt.ls,altitud[,1],altitud[,2])
alts.geo<-alt.geo
alts.geo$data<-alt.sin
plot.geodata(alts.geo)
```

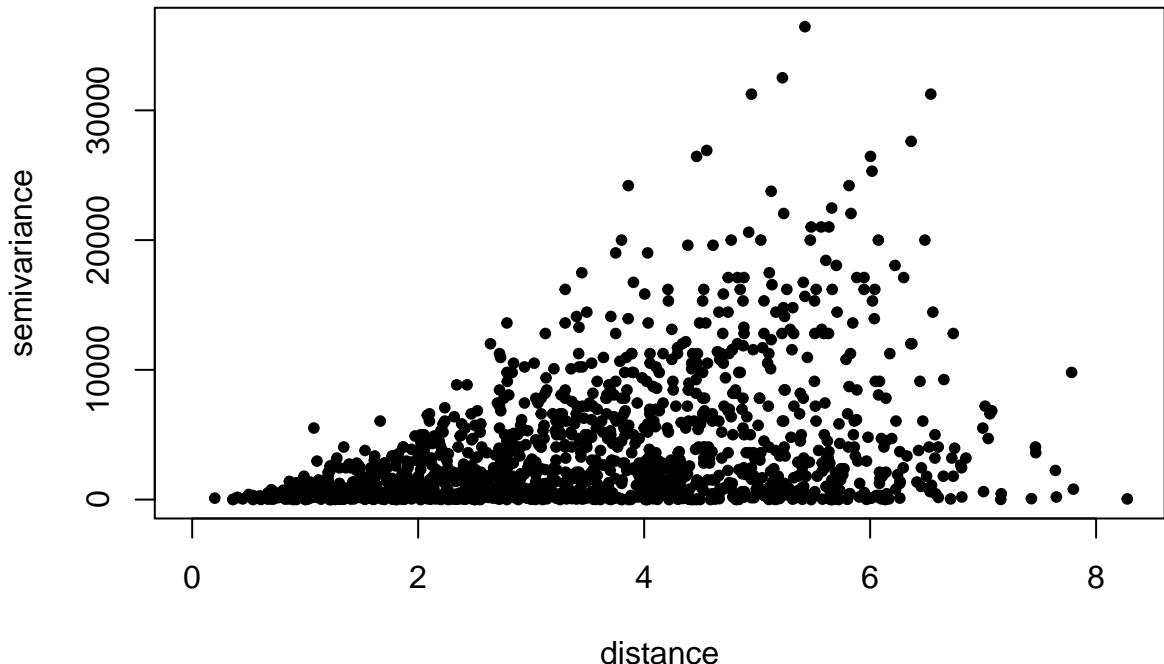


En este panel, en el que se ofrecen distintos gráficos de los residuos tras substraer la tendencia, se puede apreciar como esta efectivamente a desaparecido. También es reseñable el último gráfico en el que se puede apreciar como el rango de valores de la variable dependiente se ha reducido, principalmente porque ahora corresponden a los residuos y no a la variable altitud propiamente dicha. Los residuos se encuentran en el rango (-40, 80).

3. Estimación del variograma.

El variograma de un proceso espacial estacionario es una función capaz de expresar la diferencia que hay entre dos variables aleatorias en función de la distancia que las separa. Este puede ser obtenido mediante la función variog() del paquete geoR

```
alt.cl1 <- variog(alt.geo, option = "cloud", messages = F)
plot(alt.cl1, pch= 20)
```

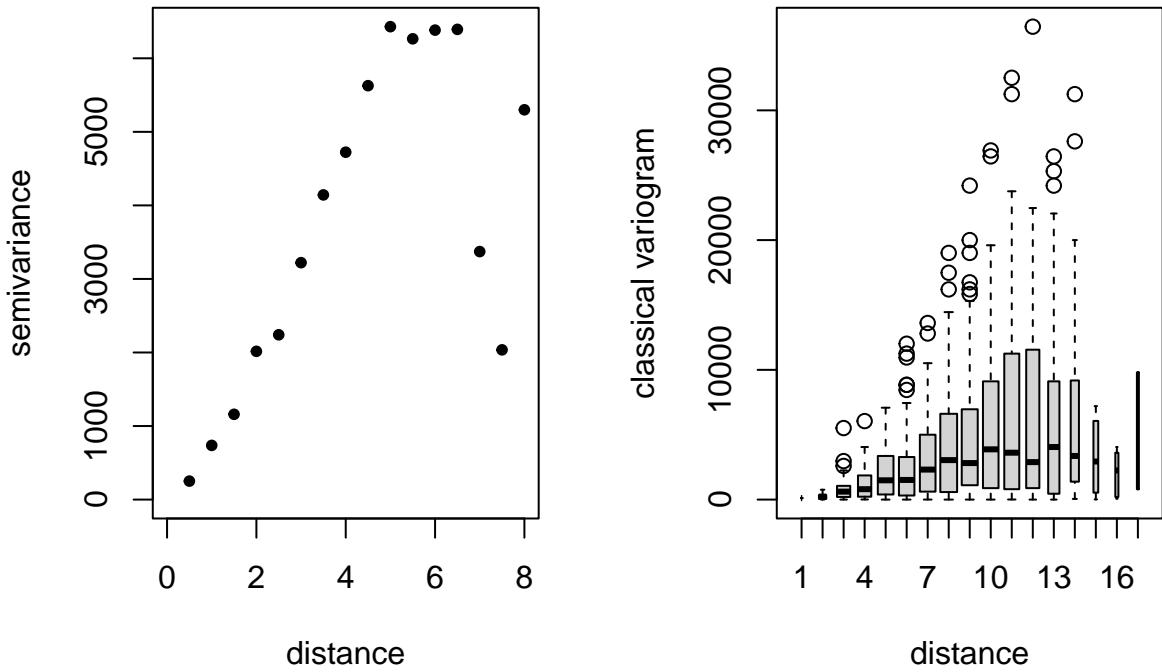


A la luz, de este gráfico, se puede concluir que los datos no son estacionarios, pues para un mismo valor de distancia existe un amplio rango de semivarianzas. Así debe de existir algún efecto sobre los datos que provoque que la varianza entre dos puntos del plano no pueda ser explicada exclusivamente con la distancia entre los puntos.

```
par(mfrow=c(1,2))
alt.v1<-variog(alt.geo,uvec=seq(0,8,0.5),messages=F)

plot(alt.v1,pch=20)

alt.vc1<-variog(alt.geo,uvec=seq(0,8,0.5),bin.cloud=T,messages=F)
# el diagrama de cajas muestra la variabilidad de los cuadrados de las diferencias
plot(alt.vc1,bin.cloud=T)
```

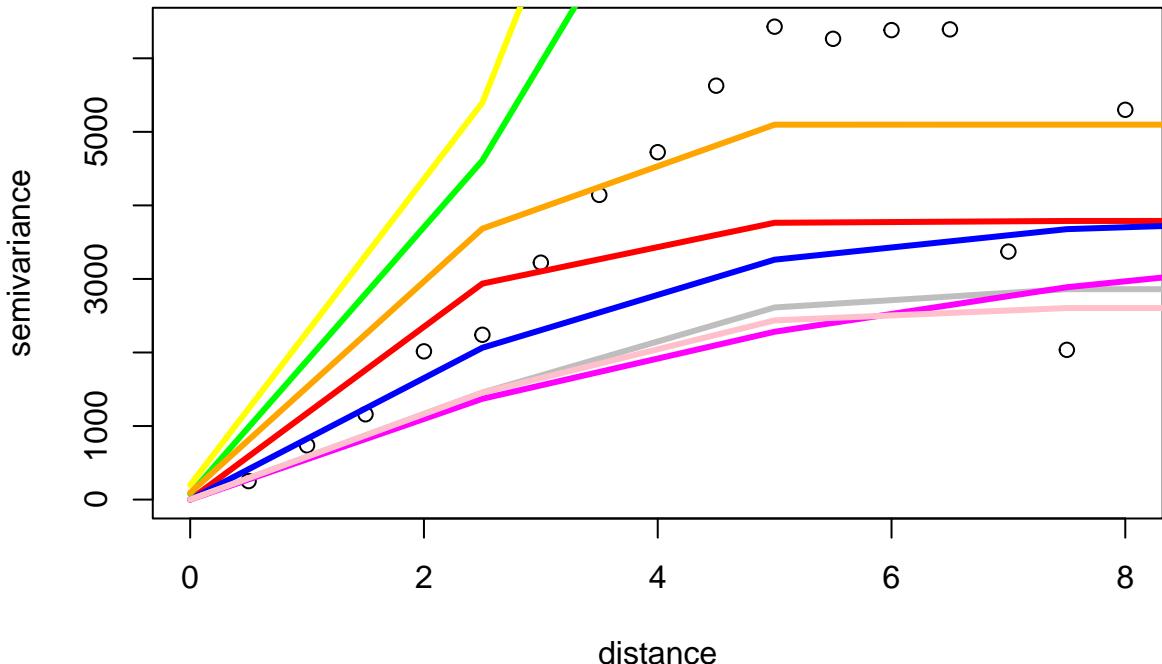


```
## pepita
alt.v1$v[1]
```

[1] 251.0893

El semivariograma empírico, es dependiente del número de intervalos que se indiquen para su cálculo, aun así las representaciones obtenidas se considerán óptimas. De este semivariograma se puede concluir que posee efecto pepita con un valor de 251.09 así como que posee un alfeizar, a la distancia de 4 U y altura de 6000. Esta estimación obtenida, no puede ser usada de forma directa para la predicción de datos. Para esto, será necesario obtener un modelo válido de semivariograma que recoja la dependencia espacial encontrada por el semivariograma empírico. Este puede ser obtenido mediante la función likfit() que posee distintos modelos de variogramas así como distintos métodos para el ajuste. La complejidad en los hiperparámetros de la función likfit() provoca que se deban de probar distintas configuraciones y mediante inspección visual escoger el más adecuado. Cabe destacar que el argumento ini, requiere de valores iniciales para el alféizar y el rango de este.

```
alt.exp.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),messages=F)
alt.sph.ml<-likfit(geodata = alt.geo, ini = c(6000, 5), cov.model="sph",messages=F)
alt.mat.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="mat",kappa=1.5,messages=F)
alt.mat2.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="mat",kappa=1,fix.nugget=T,messages=F)
alt.cir.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="cir",messages=F)
#alt.gau.ml<-likfit(geodata = alt.geo, ini = c(500000, 50),cov.model="gau")
alt.cub.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="cub",messages=F)
alt.pow.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="powered.exponential",kappa=1.75,mes
alt.pow2.ml<-likfit(geodata = alt.geo, ini = c(6000, 4),cov.model="powered.exponential",kappa=1.75,fix.
plot(alt.v1)
lines(alt.pow2.ml,max.dist=250,lwd=3,col='red')
lines(alt.mat2.ml,max.dist=250,lwd=3,col='blue')
lines(alt.pow.ml,max.dist=250,lwd=3,col='green')
lines(alt.mat.ml,max.dist=250,lwd=3,col='yellow')
lines(alt.cub.ml,max.dist=250,lwd=3,col='orange')
#lines(alt.gau.ml,max.dist=250,lwd=3,col='cyan')
lines(alt.cir.ml,max.dist=250,lwd=3,col='grey')
lines(alt.exp.ml,max.dist=250,lwd=3,col='magenta')
lines(alt.sph.ml,max.dist=250,lwd=3,col='pink')
```



De esta primera aproximación, se puede observar como la linea naranja y roja parecen ajustar de forma correcta los datos, estas se corresponde con un modelo de tipo cubico y exponencial. Ahora se probará con distintos métodos de estimación de los parámetros del modelo, para cada modelo aceptado como válido.

```

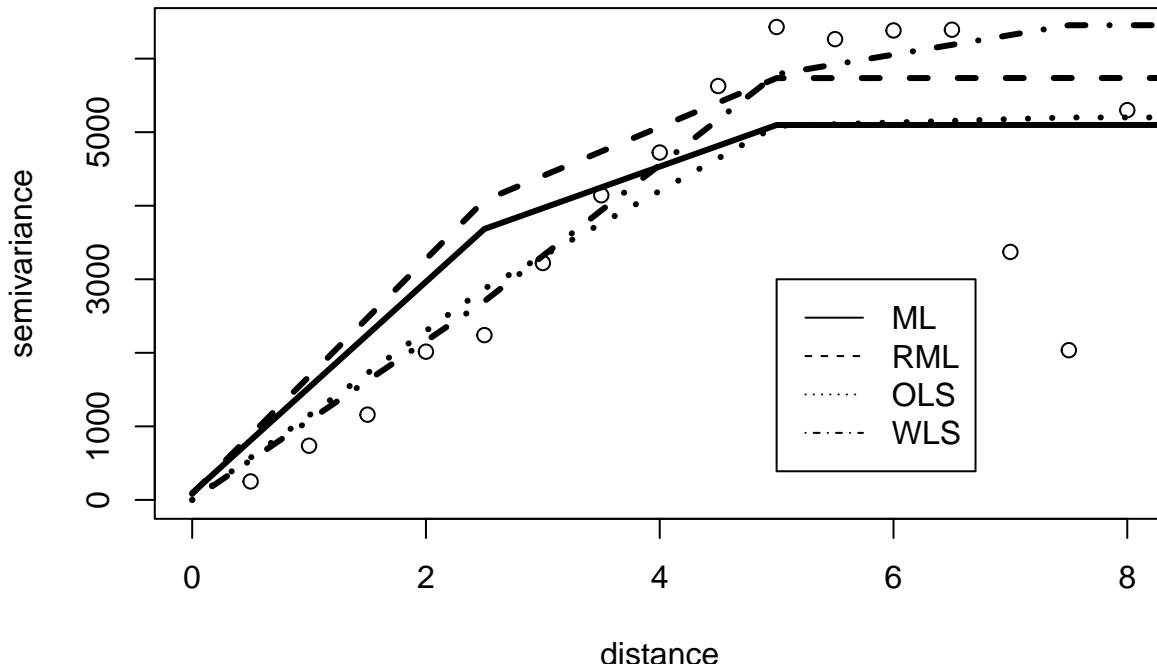
alt.cub.ml<-likfit(geodata = alt.geo, ini = c(6000, 5), cov.model="cub", kappa=1.5, messages=F)
alt.cub.rml<-likfit(geodata = alt.geo, ini = c(6000, 5), cov.model="cub", kappa=1.5, method='RML', messages=F)

## Warning in likfit(geodata = alt.geo, ini = c(6000, 5), cov.model = "cub", :
## argument "method" has changed and is now used as an argument to be passed to
## optim(). Use "lik.method" to define the likelihood method

alt.cub.ols<-variofit(vario = alt.v1, ini = c(6000, 5), cov.model="cub", kappa=1.5, weights="equal", minimi...
alt.cub.wls<-variofit(vario = alt.v1, ini = c(6000, 5), cov.model="cub", kappa=1.5, weights="npairs", messag...
plot(alt.v1, main="Modelo cúbico")
lines(alt.cub.ml,max.dist=250,lwd=3)
lines(alt.cub.rml,max.dist=250,lwd=3,lty=2)
lines(alt.cub.ols,max.dist=250,lwd=3,lty=3)
lines(alt.cub.wls,max.dist=250,lwd=3,lty=4)
legend(5,3000,legend=c('ML','RML','OLS','WLS'),lty=c(1,2,3,4))

```

Modelo cúbico

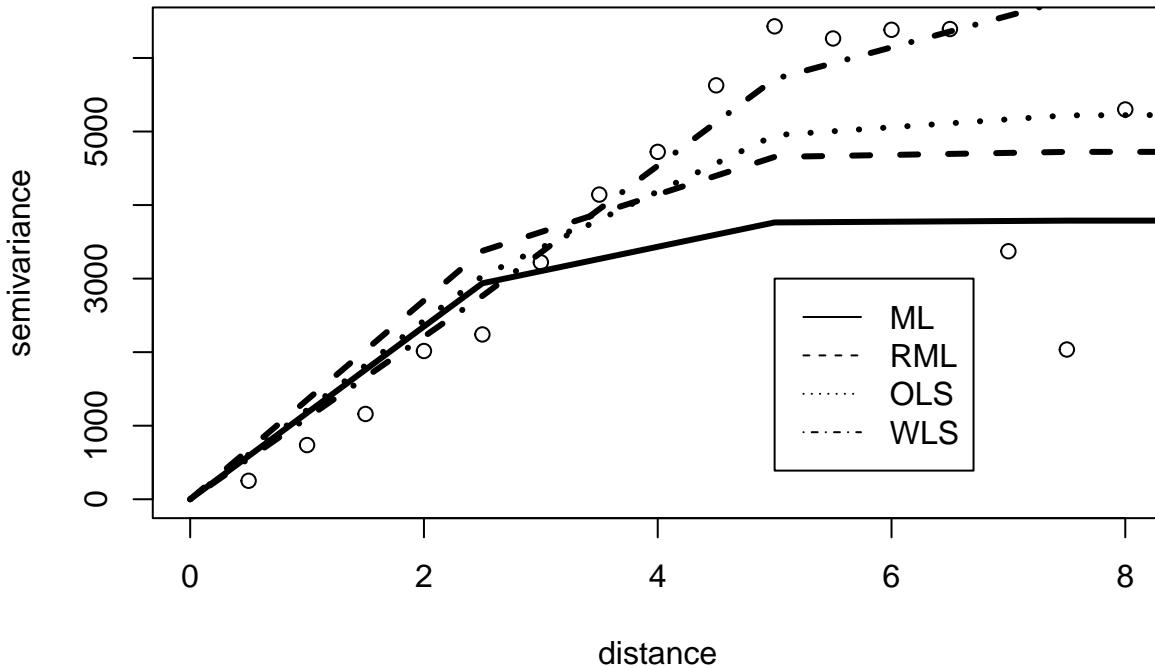


```

alt.pow2.ml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="powered.exponential",kappa=1.75,fix.=TRUE)
alt.pow2.rml<-likfit(geodata = alt.geo, ini = c(6000, 5),cov.model="powered.exponential",kappa=1.75,method="ML")
## Warning in likfit(geodata = alt.geo, ini = c(6000, 5), cov.model =
## "powered.exponential", : argument "method" has changed and is now used as an
## argument to be passed to optim(). Use "lik.method" to define the likelihood
## method
alt.pow2.ols<-variofit(vario = alt.v1, ini = c(6000, 5),cov.model="powered.exponential",kappa=1.75,weights=TRUE)
alt.pow2.wls<-variofit(vario = alt.v1, ini = c(6000, 5),cov.model="powered.exponential",kappa=1.75,weights=TRUE)
plot(alt.v1, main="Modelo Exponencial")
lines(alt.pow2.ml,max.dist=250,lwd=3)
lines(alt.pow2.rml,max.dist=250,lwd=3,lty=2)
lines(alt.pow2.ols,max.dist=250,lwd=3,lty=3)
lines(alt.pow2.wls,max.dist=250,lwd=3,lty=4)
legend(5,3000,legend=c('ML','RML','OLS','WLS'),lty=c(1,2,3,4))

```

Modelo Exponencial

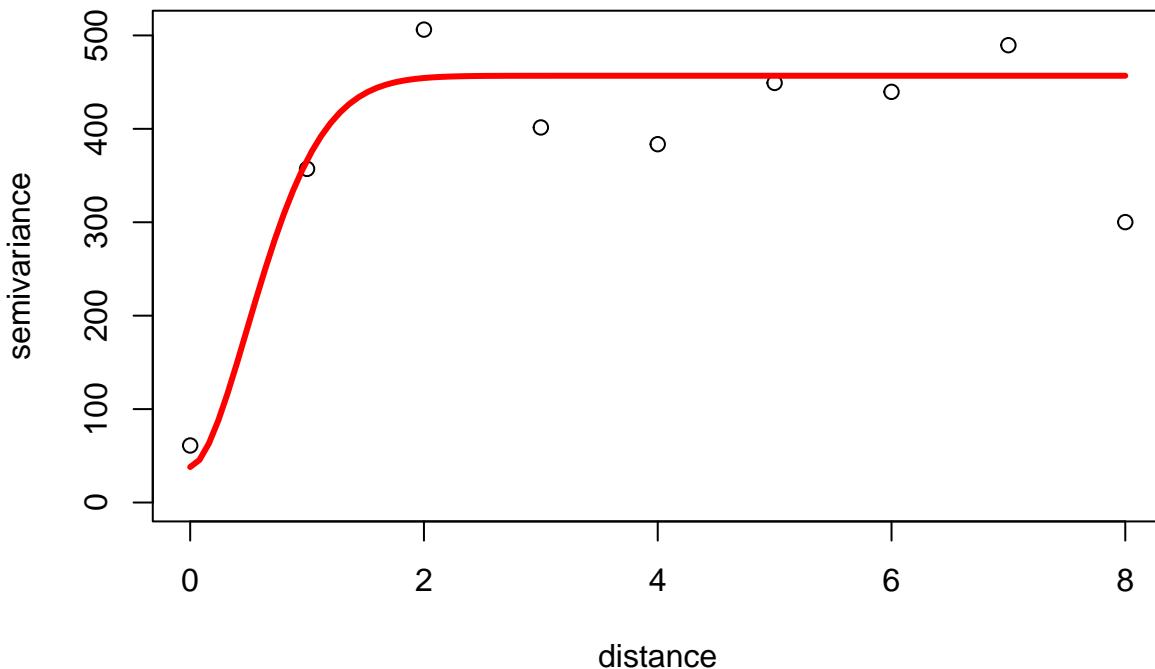


Así se puede concluir que el mejor modelo es el de tipo exponencial, con pepita fija y ajustado mediante mínimos cuadrados generalizados. Este proceso realizado para los datos originales, también se ha de repetir para el residuo de los datos tras la sustracción de la tendencia. Para no alargar más de lo necesario este trabajo, solo se mostrará el variograma obtenido siguiendo la misma rutina que para los datos original

```
alts.v1<-variog(alts.geo,uvec=seq(0,8,1),messages=F)
```

```
alts.pow.ml<-likfit(geodata = alts.geo, ini = c(500, 2),cov.model="powered.exponential",kappa=1.75, messages=F)
```

```
plot(alts.v1)
lines(alts.pow.ml,max.dist=8,lwd=3,col='red')
```



```
alts.v1$v[1]
```

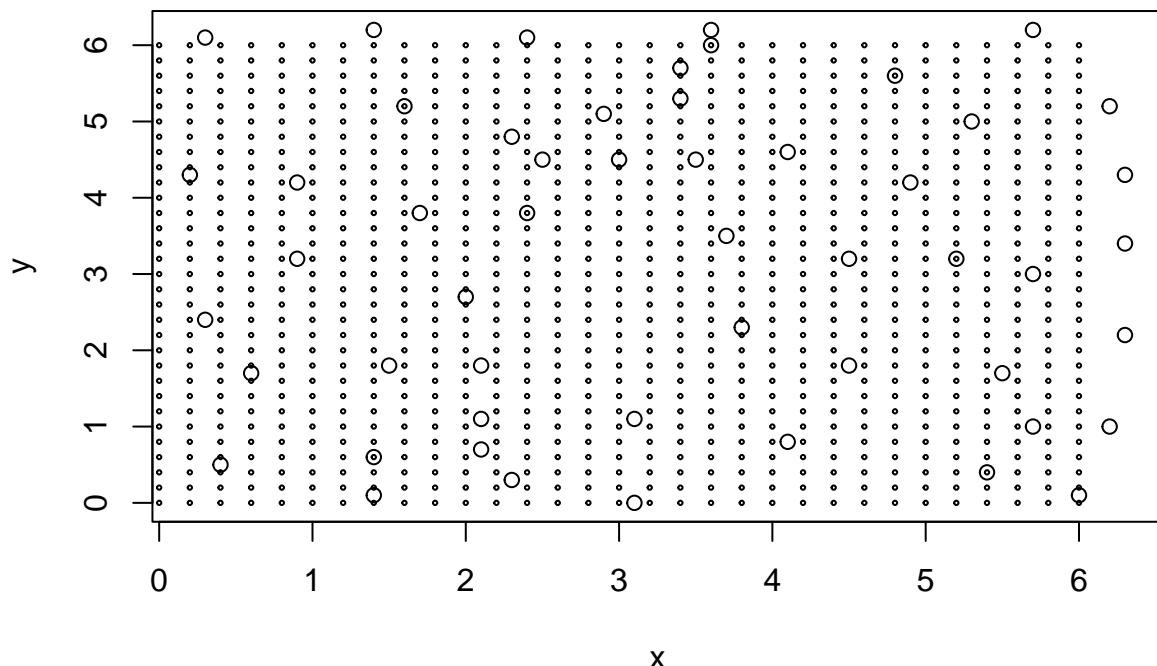
```
## [1] 61.03518
```

Este variograma, posee un alfeizar alrededor de una distancia de 2 U con un valor de semivarianza de 400 así como una pepita con un valor aproximado de 61.03.

4. Predicción espacial con kriging

A continuación se utilizará la técnica de kriging ordinario para la predicción de los datos. Este se realizará con el variograma alt.pow2.wls.

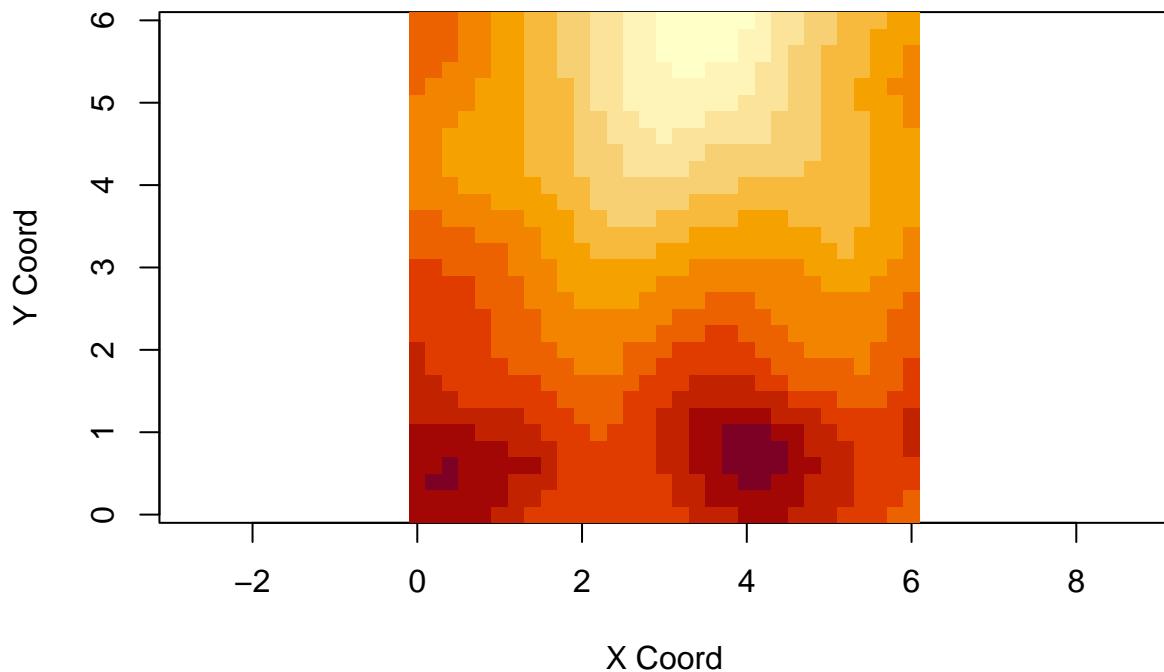
```
loci <- expand.grid(seq(0,6,l=31), seq(0,6,l=31))
plot(alt.geo$coords)
points(loci,cex=0.3)
```



```
kc1<-krige.conv(alt.geo,locations=loci,krige=krige.control(
cov.pars=alt.pow2.wls$cov.pars,nugget=alt.pow2.wls$nugget))
```

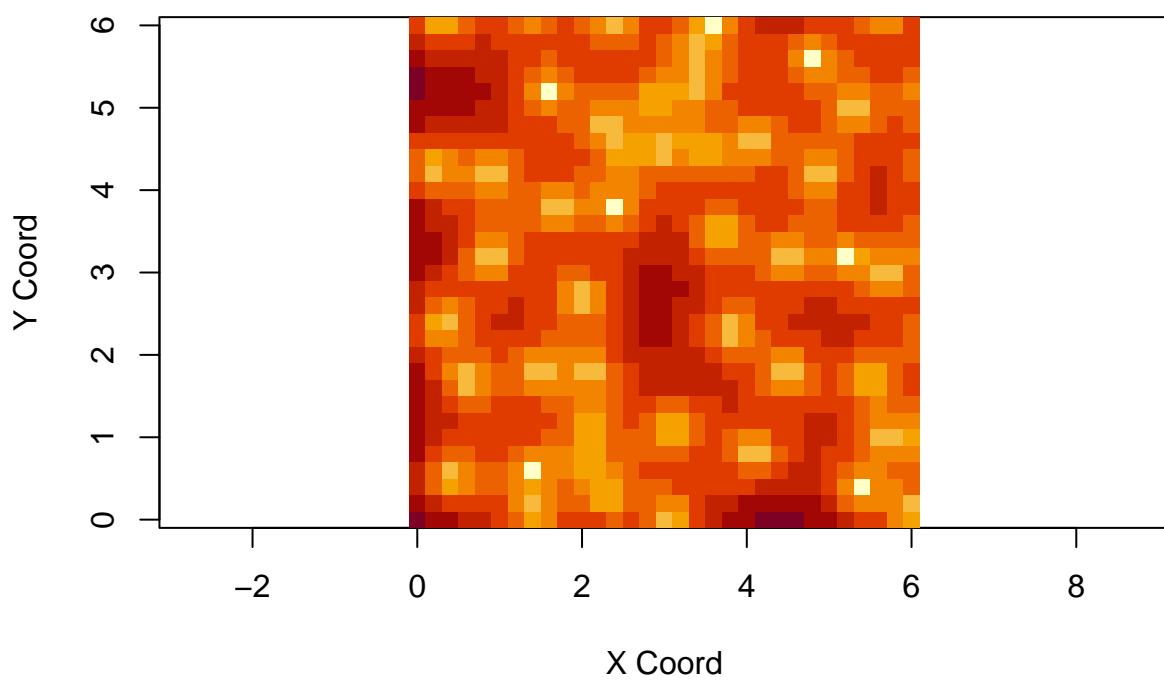
```
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
image(kc1, loc=loci, main = "estimación kriging")
```

estimación kriging



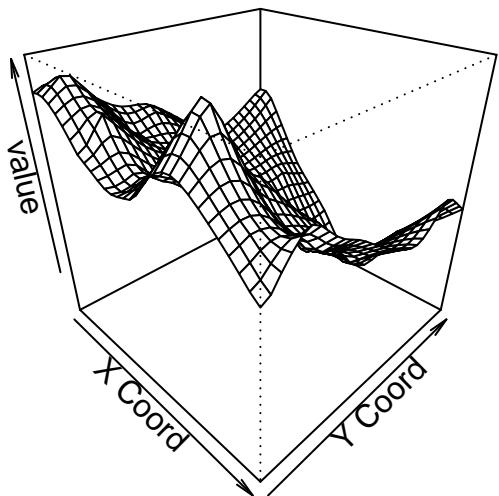
```
image(kc1, loc=loci, val=sqrt(kc1$krige.var),  
      main='error estandar')
```

error estandar



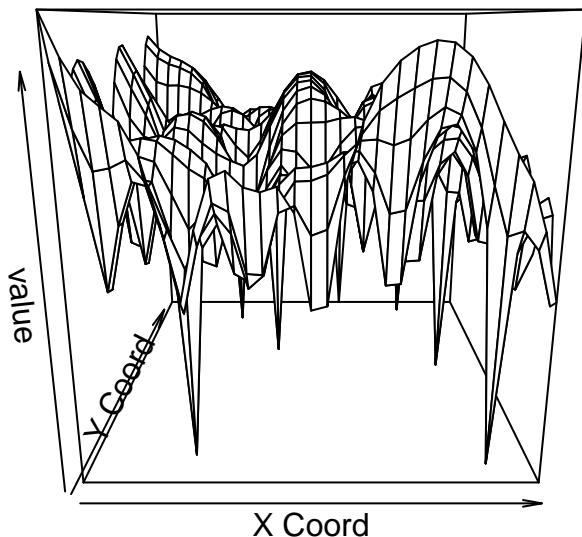
```
persp(kc1,loc=loci,main='estimacion kriging',phi=30,theta=45)
```

estimacion kriging



```
persp(kc1,loc=loci,val=sqrt(kc1$krige.var),  
      main='error estandar')
```

error estandar



El mapa de los errores permite comprobar que no existen zonas en las que exista un aumento del error de forma sistemática.

5. Predicción kriging de los residuos sin tendencia.

En este caso, se realizará la predicción sobre los datos con la tendencia eliminada. Así se usará el variograma ajustado para los residuos de los datos, alts.pow.ml. Tras la predicción sobre los residuos con la metodología de kriging se procederá a sumar la tendencia.

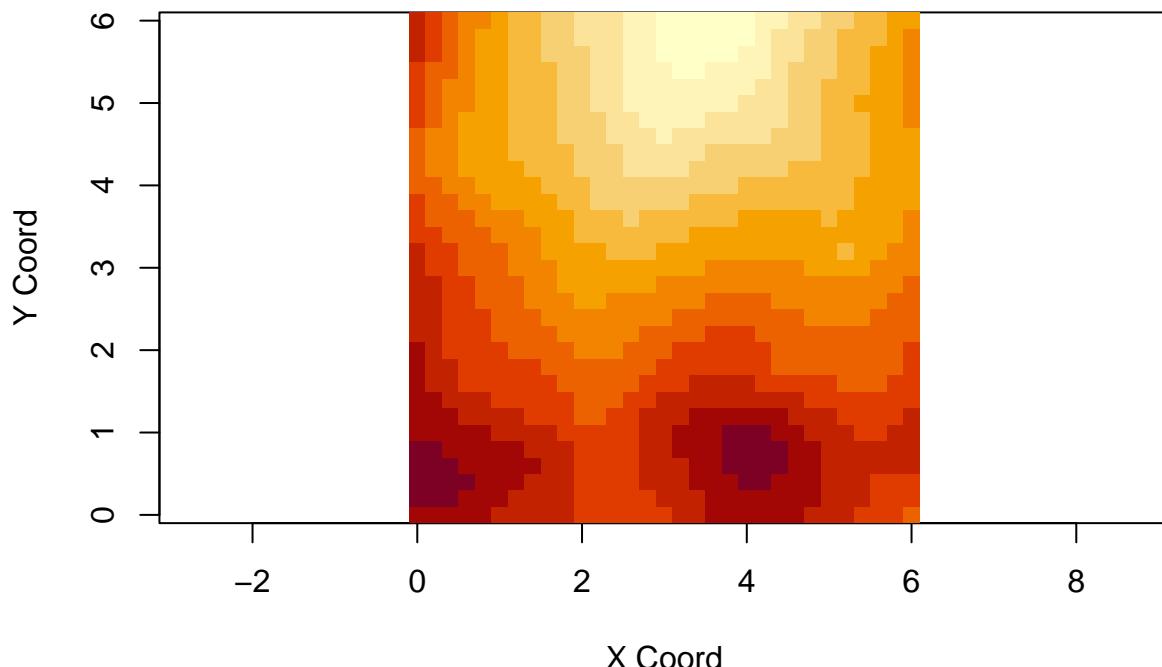
```

evaltend<-function(superf,puntos){
  predict(superf,puntos[,1],puntos[,2])}
kc2<-krige.conv(alts.geo,locations=loci,krige=krige.control(
  cov.pars=alts.pow.ml$cov.pars,nugget=alts.pow.ml$nugget))

## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
image(kc2,loc=loci,val=kc2$predict+evaltend(alt.ls,loci),
  main='estimacion kriging')

```

estimacion kriging

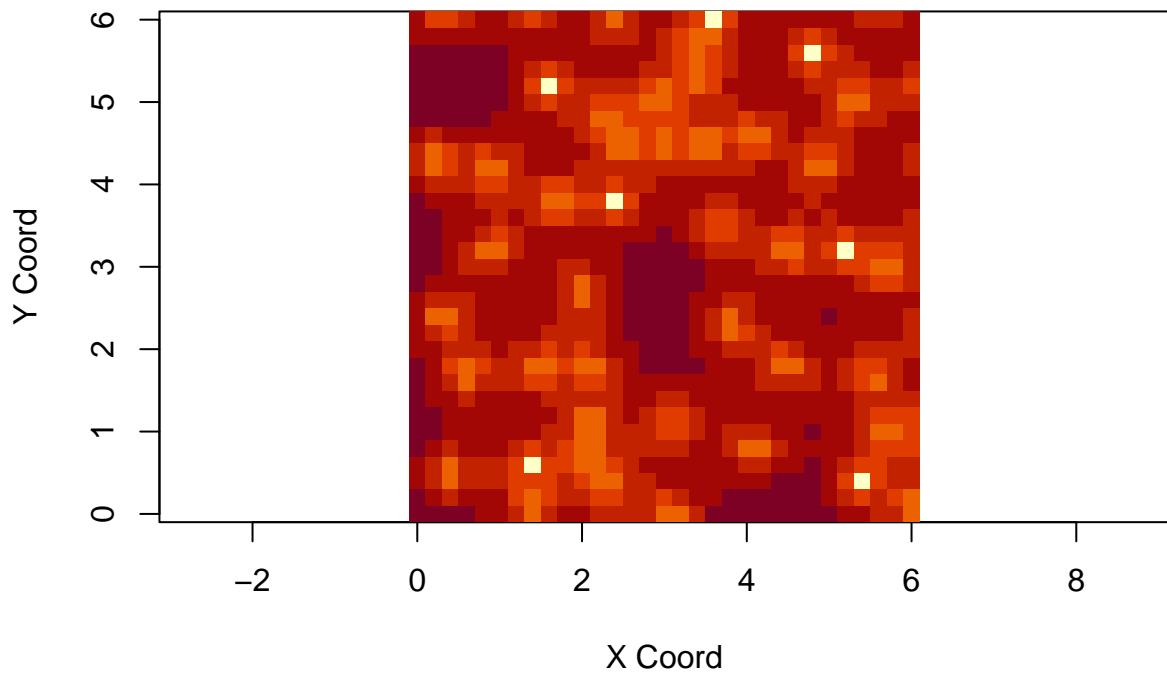


```

image(kc2,loc=loci,val=sqrt(kc2$krige.var),
  main='error estandar')

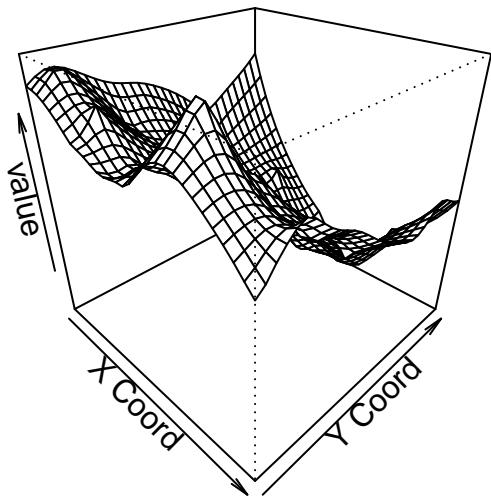
```

error estandar



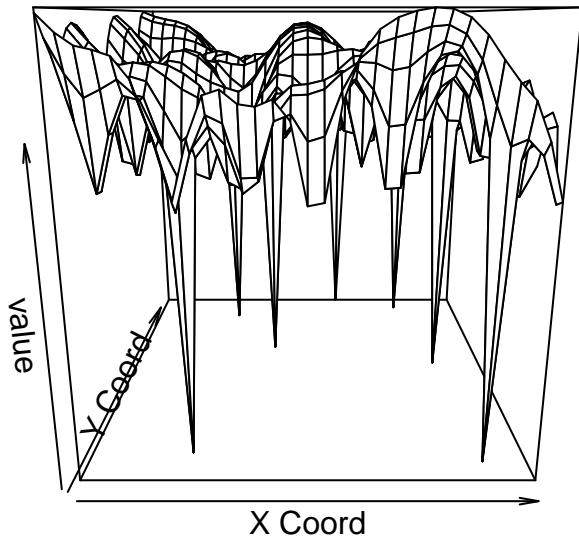
```
persp(kc2, loc=loci, val=kc2$predict+evaltend(alt.ls,loci),  
      main='estimacion kriging', phi=30, theta=45)
```

estimacion kriging



```
persp(kc2, loc=loci, val=sqrt(kc2$krige.var),  
      main='error estandar')
```

error estandar



```
mean(sqrt(kc2$krige.var))
```

```
## [1] 16.16779
```

```
mean(sqrt(kc1$krige.var))
```

```
## [1] 29.09022
```

Se puede observar como en esta metodología el error es ligeramente menor que en el kriging ordinario realizado anteriormente.

6. Predicción espacial con Kriging universal.

En esta ocasión se realizará la metodología de kriging universal que incorpora directamente la superficie de tendencia así como otras funciones ligadas a las localizaciones, evaluadas en los puntos de observación y en los de predicción.

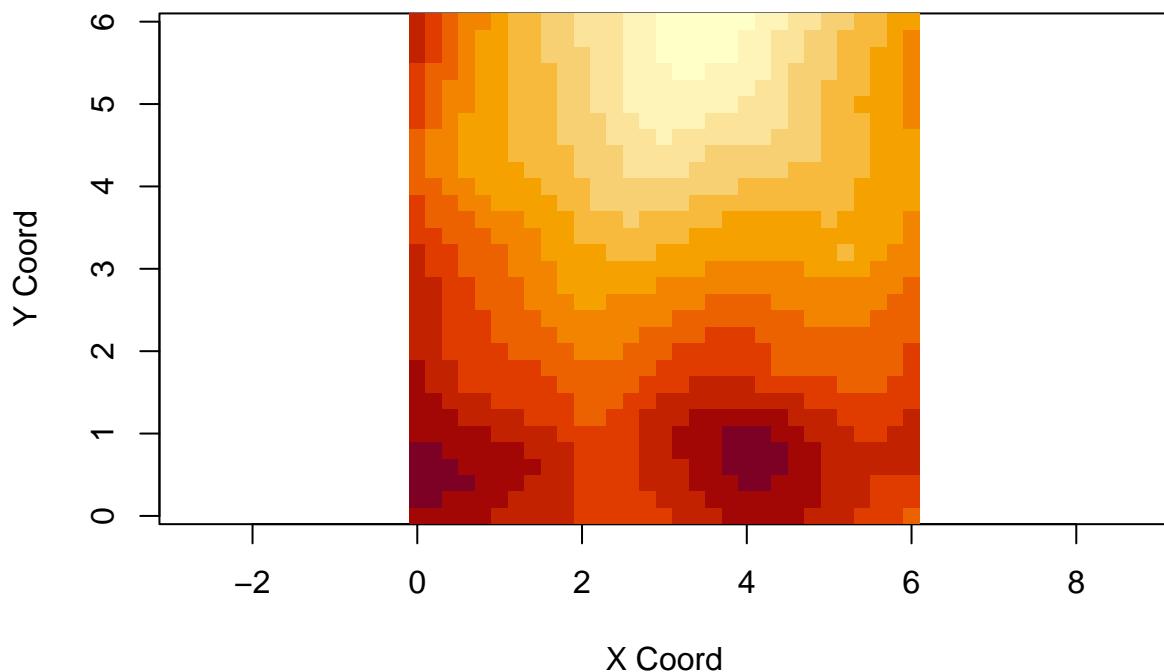
```
kc3<-krige.conv(alt.geo,locations=loci,krige=krige.control(trend.d=~evaltend(alt.ls,alt.geo$coords),tre
```

```
## krige.conv: model with mean defined by covariates provided by the user
```

```
## krige.conv: Kriging performed using global neighbourhood
```

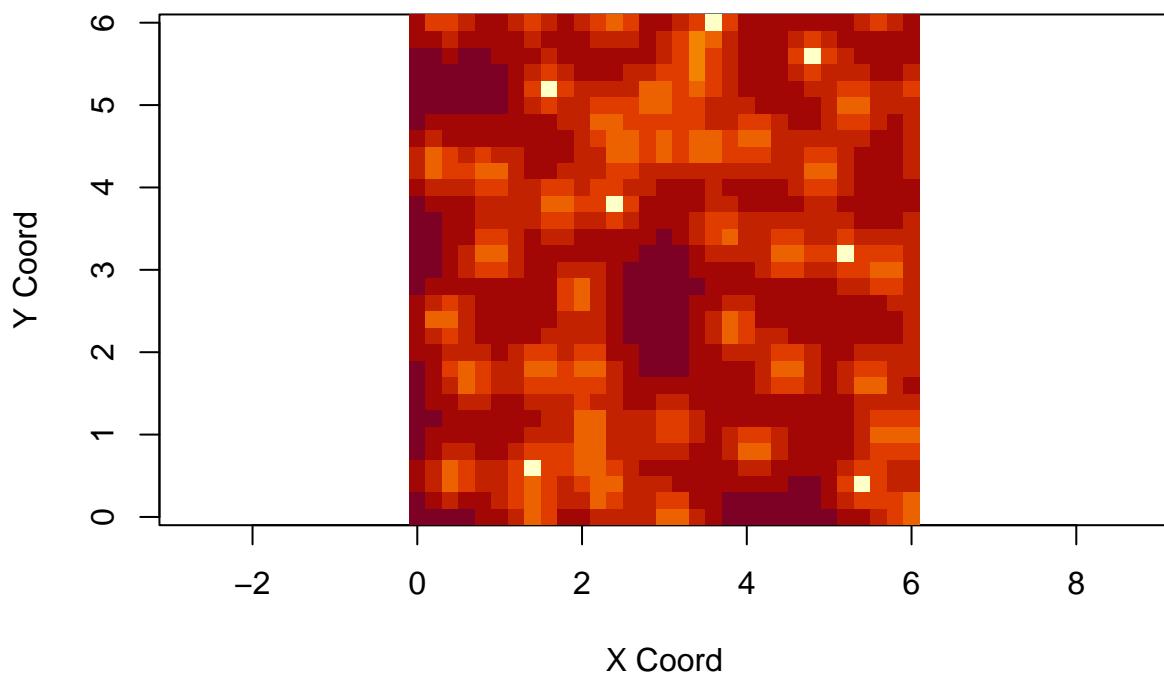
```
image(kc3,loc=loci,main='estimacion kriging')
```

estimacion kriging



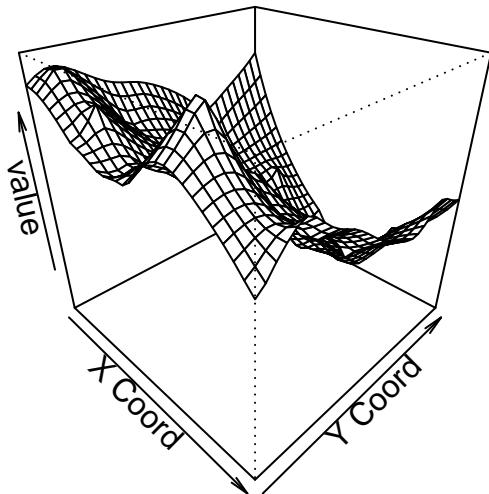
```
image(kc3, loc=loci, val=sqrt(kc3$krige.var),  
      main='error estandar')
```

error estandar



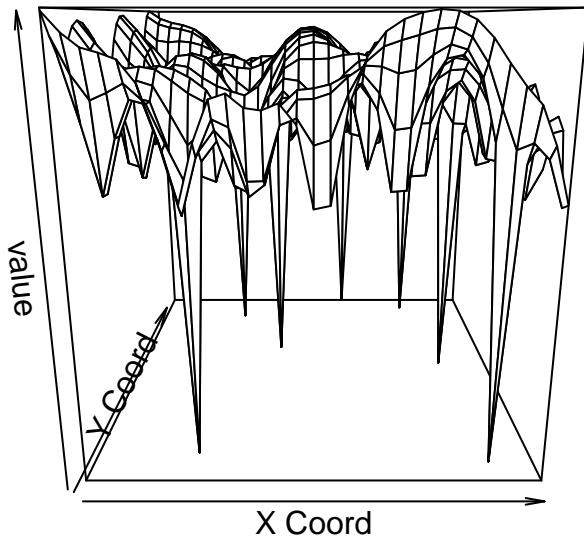
```
persp(kc3,loc=loci,main='estimacion kriging',phi=30,theta=45)
```

estimacion kriging



```
persp(kc3,loc=loci,val=sqrt(kc3$krige.var),  
      main='error estandar')
```

error estandar



```
mean(sqrt(kc3$krige.var))
```

```
## [1] 16.19231
```

```
mean(sqrt(kc2$krige.var))
```

```
## [1] 16.16779
```

```
mean(sqrt(kc1$krige.var))
```

```
## [1] 29.09022
```

En este caso, el error medio ses similar al obtenido para el kriging ordinario sobre los datos originales.

7. Comparación de las predicciones.

Por último se evaluarán las predicciones obtenidas por los tres métodos descritos anteriormente. Las predicciones se han realizado para la coordendad ($x = 3$ U, $y = U$) que se corresponde con el loci[481,]

```
predi1<-function(i)
  c(kc1$predict[i]-1.96*sqrt(kc1$krige.var[i]),kc1$predict[i],
    kc1$predict[i]+1.96*sqrt(kc1$krige.var[i]))
predi1(481)
```

```
## [1] 745.1279 819.4280 893.7281
```

```
predi2<-function(i)
  c(kc2$predict[i]+evaltend(alt.ls,loci[i,])-1.96*sqrt(kc2$krige.var[i]),kc2$predict[i]+evaltend(alt.ls,
    kc2$predict[i]+evaltend(alt.ls,loci[i,])+1.96*sqrt(kc2$krige.var[i]))
predi2(481)
```

```
## [1] 779.3394 816.9058 854.4722
```

```
predi3<-function(i)
  c(kc3$predict[i]-1.96*sqrt(kc3$krige.var[i]),kc3$predict[i],
    kc3$predict[i]+1.96*sqrt(kc3$krige.var[i]))
predi3(481)
```

```
## [1] 779.4070 816.9916 854.5761
```

Todas las predicciones arrojan un intervalo de confianza al 95 % lógico si se atienden a los datos originales. Cabe destacar que la aproximación kriging sobre los residuos sin tendencia ofrece el intervalo de confianza de menor longitud. Siendo el kriging universal el que ofrece el segundo intervalo de confianza de menor longitud. Así nos quedaríamos con el método de kriging ordinario sobre los residuos como método más optimo para la predicción.