

```

# Cargamos la librería que necesitamos y comprobamos sus ficheros de datos de ejemplo
library(gstat)
data(package="gstat")

#Seleccionamos un fichero de datos para realizar el estudio. Para este ejemplo he utilizado
# el fichero meuse

data(meuse)
par(mfrow=c(2,2))
boxplot(meuse$cadmium,main='Cadmio')
boxplot(meuse$copper,main='Cobre')
boxplot(meuse$lead,main='Plomo')
boxplot(meuse$zinc,main='Zinc')

# Comprobamos la normalidad de los datos
shapiro.test(meuse$cadmium)
shapiro.test(meuse$copper)
shapiro.test(meuse$lead)
shapiro.test(meuse$zinc)

# Escogemos una variable para estudiar. Vamos a continuar el análisis con la variable cobre
# Comenzamos el análisis estructural comprando la existencia de tendencia (función m(x))
Regresion<-lm(copper~x+y, data=meuse)
summary(Regresion)

# La representación gráfica de la nube de semivariograma nos da una idea de la variabilidad de los datos.
# Representamos la nube de semivariograma para el estimador por el método de los momentos y la nube
# de semivariograma para el estimador robusto. Fijaos en el cambio de escala del eje Y.
# Sería importante estudiar las opciones de la función variogram mediante un help(variogram).

help(variogram)
plot(variogram(copper~1,~x+y,meuse,ressie = F, cloud = T),main="Estimador por el método de los
momentos",sub="Nube de semivariograma")
windows()
plot(variogram(copper~1,~x+y,meuse,ressie = T, cloud = T),main="Estimador robusto",sub="Nube de
semivariograma")

# Hasta aquí podemos constatar: presencia de tendencia y falta de normalidad en los datos.

# Cálculo de semivariogramas experimentales. Semivariograma por el método de los momentos y
# semivariograma robusto. Este último es más apropiado en el caso de no normalidad y/o la falta
# de simetría en los datos

plot(variogram(copper~x+y,~x+y,meuse,ressie = F),main="Estimador por el método de los momentos",
sub="Semivariograma experimental")
windows()
plot(variogram(copper~x+y,~x+y,meuse,ressie = T),main="Estimador robusto",sub="Semivariograma
experimental")

# Cálculo de superficies de semivariogramas para identificar anisotropías
VgExp <-variogram(copper~x+y,~x+y,meuse,ressie = T, cutoff=1500,width = 100,map=T)
plot(VgExp,main="Datos de cobre",sub="Superficie de semivariograma")

# La imagen anterior muestra una anisotropía geométrica. La dependencia de mayor alcance está en
# la dirección 50 grados y la de menor alcance en 140
#Podemos calcular los semivariogramas experimentales en esas direcciones
VgExp <- variogram(copper~x+y,~x+y,meuse,ressie = T, cutoff=2000,alpha=c(5,50,95,140))
plot(VgExp,xlab="Distancia |h|",ylab="Semivariograma experimental")

# Ajuste de semivariogramas teóricos

```

```

# Comenzamos mostrando varios modelos de semivariogramas teóricos.
# Puede resultar interesante ver los comentarios incluidos en los helps
# de las dos funciones
vgm()
show.vgms()
# Pasamos ahora a ajustar semivariogramas teóricos a semivariogramas experimentales.

# A la vista de los semivariogramas experimentales debemos decidirnos por un modelo
# teórico para ajustar.
# El ratio de alcances es el único parámetro que debe ser ajustado 'a ojo' viendo el gráfico del ajuste.
# Comenzamos por un modelo esférico.
Sph1<-fit.variogram(VgExp,vgm(350,'Sph',1000,100,anis=c(50,0.6)))
plot(VgExp,model=Sph1)
Sph1
attr(Sph1,'SSErr')

# La suma de errores al cuadrado, SSE, es una referencia de la bondad del ajuste
# Podemos ir variando los parámetros iniciales, el ratio de anisotropía
# y las ponderaciones para buscar el mejor ajuste. Las posibles ponderaciones se muestran en la tabla 2
# incluida al final de este fichero. Por defecto se asume fit.method = 7

Sph2<-fit.variogram(VgExp,vgm(350,'Sph',1000,100,anis=c(50,0.6)),fit.method = 2)
plot(VgExp,model=Sph2)
Sph2
attr(Sph2,'SSErr')

# Continuamos por un modelo exponencial.
Exp1<-fit.variogram(VgExp,vgm(350,'Exp',400,120,anis=c(50,0.6)))
Exp1
plot(VgExp,model=Exp1)
attr(Exp1,'SSErr')

Exp2<-fit.variogram(VgExp,vgm(350,'Exp',400,120,anis=c(50,0.6)),fit.method = 2)
Exp2
plot(VgExp,model=Exp2)
attr(Exp2,'SSErr')

# Finalizamos con un modelo gaussiano.
Gau1<-fit.variogram(VgExp,vgm(350,'Gau',400,120,anis=c(50,0.6)))
Gau1
plot(VgExp,model=Gau1)
attr(Gau1,'SSErr')

Gau2<-fit.variogram(VgExp,vgm(350,'Gau',400,120,anis=c(50,0.6)),fit.method = 2)
Gau2
plot(VgExp,model=Gau2)
attr(Gau2,'SSErr')

# También puede ser interesante explorar la posibilidad de utilizar modelos teóricos anidados
mdlteorico <- vgm( 300, "Sph", 1000, add.to = vgm(1, "Exp", 400, nugget = 100), anis=c(50,0.6))
ModAnid<-fit.variogram(VgExp, mdlteorico)
ModAnid
plot(VgExp,model= ModAnid)
attr(ModAnid,'SSErr')
# Aunque, en este caso del cobre,el modelo exponencial parece sobrar

```

Modelo	Meseta parcial	Rango	Ratio de anisotropía	Pepita	SSE	Fit	Rango efectivo
Esférico	344,84551	1569,673	0,6	84,42059	215,2519	7	1273,78964
	339,51125	1518,035	0,6	82,74302	1550,322	2	1231,885403
Exponencial	417,92493	796,1874	0,6	57,01806	218,5828	7	2388,5622
	426,869916	546,5685	0,6	6,891522	1447,968	2	1639,7055
Gausiano	288,6591	667,9951	0,6	122,1929	218,8591	7	1157,001452
	276,1322	783,1529	0,6	142,5199	1427,380	2	1356,460613

Tabla 1. Tabla de resumen de los ajustes obtenidos

fit	fit by	weight
0	-	- (no fit)
1	gstat	N_j
2	gstat	$N_j / \{\gamma(h_j)\}^2$
3	gnuplot	N_j
4	gnuplot	$N_j / \{\gamma(h_j)\}^2$
5	gstat	REML
6	gstat	no weights (OLS)
7	gstat	N_j / h_j^2

Tabla 2. Posibles ponderaciones para el ajuste del semivariograma experimental