

```
library(gstat)
library(sp)
data(meuse)

#Multivariate variogram modelling
data(meuse)
shapiro.test(meuse$cadmium)
shapiro.test(log(meuse$cadmium))

coordinates(meuse) <- ~ x+y

bubble(meuse, "cadmium")
bubble(meuse, "copper")
bubble(meuse, "lead")
bubble(meuse, "zinc")

#Combine the four variables
g <- gstat(NULL, "logCd", log(cadmium) ~ 1, meuse)
g <- gstat(g, id="logCu", form=log(copper) ~ 1, data=meuse)
g <- gstat(g, "logPb", log(lead) ~ 1, meuse)
g <- gstat(g, "logZn", log(zinc) ~ 1, meuse)
g

vm <- variogram(g)

#You only need to specify one model. In fact, there are 10 semivariogram models

vm.fit <- fit.lmc(vm, g, vgm(1, "Sph", 800, 1))
plot(vm, vm.fit)

#cokriging
data(meuse.grid)
coordinates(meuse.grid) <- ~x+y
cok.maps <- predict(vm.fit, meuse.grid)

spplot(cok.maps["logCd.pred"])

#stratified kriging

library(gstat)
library(sp)
data(meuse)
data(meuse.grid)
coordinates(meuse) <- ~x+y
coordinates(meuse.grid) <- ~x+y
gridded(meuse.grid) <- TRUE

#add the strata variable for meuse data
```

```

meuse$part.a <- idw(part.a ~ 1, meuse.grid, meuse, nmax=1)$var1.pred

#Semivariogram modeling for each stratum independently
var1 <- variogram(log(zinc) ~ 1, meuse[meuse$part.a==0,])
plot(var1)
vgm1 <- vgm(0.5, "Gau", 400, 0.1)
var1.fit <- fit.variogram(var1, vgm1)

var2 <- variogram(log(zinc) ~ 1, meuse[meuse$part.a==1,])
plot(var2)
vgm2 <- vgm(0.6, "Sph", 600)
var2.fit <- fit.variogram(var2, vgm2)

#Kriging for each stratum independently
x1 <- krige(log(zinc) ~ 1, meuse[meuse$part.a==0,],
  meuse.grid[meuse.grid$part.a==0,], model=var1.fit,
  nmin=20, nmax=40, maxdist=1400)

x2 <- krige(log(zinc) ~ 1, meuse[meuse$part.a==1,],
  meuse.grid[meuse.grid$part.a==1,], model=var2.fit,
  nmin=20, nmax=40, maxdist=1000)

#Combine the kriging results. Pay attention to the non-smooth part at the edge that
separates the two strata

lz.stk <- rbind(as.data.frame(x1),as.data.frame(x2))
coordinates(lz.stk) <- ~x+y
spplot(lz.stk["var1.pred"])

# cross validation
library(gstat)
library(sp)
data(meuse)
coordinates(meuse) <- ~x+y
sel100 <- sample(1:155,100)
m.model <- meuse[sel100,]
m.valid <- meuse[-sel100,]

var.m <- variogram(log(zinc) ~ 1, m.model)
plot(var.m)
vgm.m <- vgm(0.5, "Sph", 700, 0.1)
var.mfit <- fit.variogram(var.m, vgm.m)
plot(var.m, var.mfit)
var.mfit
m.valid.pred <- krige(log(zinc) ~ 1, m.model, m.valid, model=var.mfit)

resid.kr <- log(m.valid$zinc) - m.valid.pred$var1.pred
summary(resid.kr)

```

Class_11202018

```
resid.mean <- log(m.valid$zinc) - mean(log(m.valid$zinc))
R2 <- 1-sum(resid.kr^2) / sum(resid.mean^2)

m.valid.pred$res <- resid.kr
bubble(m.valid.pred, "res")

#To use the data to a fuller extent (leave-one-out cross validation nfold=155)

nfold <- 3
part <- sample(1:nfold, 155, replace=TRUE)

for (i in 1: nfold) {
  sel <- (part != i)
  m.model <- meuse[sel,]
  m.validate <- meuse[-sel,]
  # variogram modeling, kriging, residual
}

#a more automated way
v.fit <- vgm(0.59, "Sph", 897, 0.05)
cv155 <- krige.cv(log(zinc) ~ 1, meuse, v.fit, nfold=5)
bubble(cv155, "residual")
summary(cv155)

#stochastic simulation
v <- variogram(log(zinc) ~ 1, meuse)
v.fit <- fit.variogram(v, vgm(0.6, "Sph", 1000, 0.1))
lzn.sim <- krige(log(zinc) ~ 1, meuse, meuse.grid, v.fit, nsim = 6, nmax = 40)
splot(lzn.sim)
```