argila\_dnos

LC

4 de dezembro de 2017

# Modelagem geoestatística do conteúdo de argila da bacia do DNOS

O presente estudo tem como objetivo realizar a predição espacial do teor de argila no solo na bacia do reservatorio do Departamento Nacional de Obras de Saneamento-Companhia Riograndense de Saneamento (DNOS-CORSAN), no município de Santa Maria/RS. Os dados foram coletados em locais selecionados intencionalmente ou por conveniência, perfazendo um total de 400 observações em pontos amostrais. Tais dados fazem parte do projeto FEBR e estão disponíveis [aqui](https://drive.google.com/drive/folders/0B7xsLbrOA23oNkRKUEUtajF4Wjg). O clima da região, conforme o sistema de classificação de Köppen é subtropical do tipo Cfa. A área compreende a região de transição entre o Planalto e a Depressão Central, denominada Rebordo do Planalto do RS. As características da área de estudo condicionam uma alta variabilidade geomorfológica. O relevo local varia de plano a montanhoso. A geomorfologia da área pode ser caracterizada como complexa, sendo uma transição de rochas ígneas (região do Planalto) e rochas sedimentares (Depressão Central). Primeiramente serão instalados os pacotes necessários para o carregamento das funções que serão utilizadas no desenvolvimento de todo o trabalho. Embora sejam utilizadas em diferentes partes do trabalho, para fins práticos recomenda-se que estes sejam instalados em sua totalidade para o bom andamento das análises. Para o carregamento dos dados do FEBR, que deve ser instalado a partir do github, será utilizado o comando .

Após os pacotes serem instalados, devem ser carregados (a partir do comando ):

library(mapview)  
library(devtools)  
library(febr)  
library(magrittr)  
library(dplyr)  
library(glue)  
library(lattice)  
library(latticeExtra)  
library(georob)  
library(sp)  
library(rgdal)  
require(raster)  
library(caret)  
library(bookdown)  
library(gstat)

Devemos também definir o sistema de referência de coordenadas geográficas (Fonte: <http://spatialreference.org/ref/epsg/>) utilizadas, além da criação uma rampa de cores que serão utilizadas na geração dos mapas.

wgs84utm22s <- sp::CRS('+proj=utm +zone=22 +south +ellps=WGS84 +datum=WGS84 +units=m +no\_defs')  
sirgas2000 <- sp::CRS('+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no\_defs')  
  
col\_soil\_var <- topo.colors(100)

Com os pacotes carregados, serão baixados os dados do FEBR para a área estudada. Primeiramente serão descarregados os arquivos do mapa pedológico para a área estudada em um diretótio local. Estes dados serão utilizados para comparação dos métodos de krigagem, investigando a influência da utilização do mapa pedológico como covariável preditora.

data\_folder <- '../data/'  
ext <- c('dbf', 'prj', 'shp', 'shx')  
files <- (glue('pedologia25.{ext}'))  
download <- !all(files %in% list.files(data\_folder))  
if (download) {  
 url <- 'https://github.com/samuel-rosa/UFSM-SOL-843/tree/master/data/'  
 url <- glue('{url}{files}')  
 destfile <- glue('{data\_folder}{files}')  
 for (i in 1:length(files)) {  
 download.file(url = url[i], destfile = destfile[i])  
 }  
}

Com a intenção de demonstrar a distribuição das classes de solo e unidades de mapeamento dentro da bacia, além da distribuição dos pontos onde foram coletadas as amostras, será carregado o mapa pedologico a partir da função . Será criada também uma rampa de cores para representação das unidades de mapeamento. Também será utilizada a função para projetar as coordenadas original no plano cartesiano (UTM).

pedologia25 <-   
 glue('../data/pedologia25.shp') %>%   
 raster::shapefile(stringsAsFactors = TRUE, encoding = 'UTF-8') %>%   
 sp::spTransform(wgs84utm22s)  
  
pedologia25@data

## um  
## 0 Cambissolo - Neossolo  
## 1 Cambissolo - Neossolo  
## 2 Planossolo Háplico  
## 3 Cambissolo - Neossolo  
## 4 Cambissolo - Neossolo  
## 5 Neossolo Litólico - Neossolo Regolí­tico  
## 6 Neossolo Regolí­tico  
## 7 Argissolo Bruno-Acinzentado  
## 8 Neossolo Flúvico  
## 9 Planossolo Háplico  
## 10 Argissolo Vermelho  
## 11 Neossolo Litólico  
## 12 Neossolo Regolítico  
## 13 Cambissolo - Neossolo  
## 14 Neossolo Litólico  
## 15 Cambissolo - Neossolo  
## 16 Neossolo Litólico  
## 17 Cambissolo - Neossolo  
## 18 Argissolo Vermelho  
## 19 Planossolo Háplico  
## 20 Cambissolo - Neossolo  
## 21 Planossolo Háplico  
## 22 Cambissolo - Neossolo  
## 23 Cambissolo - Neossolo  
## 24 Cambissolo - Neossolo

col\_soil\_um <- terrain.colors(nlevels(pedologia25$um))

Visto isso, agora será descarregado da base de dados do FEBR o conjunto de dados ctb0003, referente ao conjunto de dados do solo da bacia do reservatório do DNOS. Também será criado o objeto espacial com os pontos onde foram amostrados os valores de argila no solo.

dnos\_observations <- febr::observations('ctb0003', which.cols = 'all', progress = FALSE)

## stack.obs can only be used with standard columns... switching to FALSE

## Downloading dataset ctb0003...

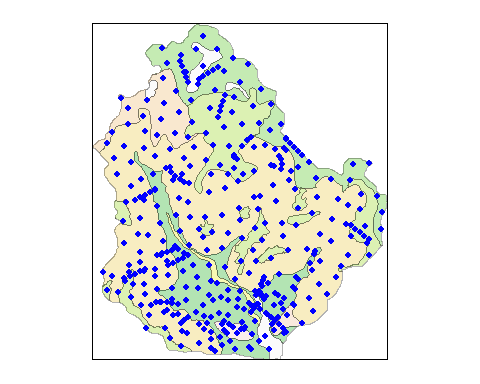
dnos\_layers <- febr::layers('ctb0003', soil.vars = "argila", which.cols = 'all', missing.data = 'keep', progress = FALSE)

## stack.datasets can only be used with standard columns... switching to FALSE  
## Downloading dataset ctb0003...

id <- c('dataset\_id', 'observacao\_id')  
dnos <-   
 merge(dnos\_observations, dnos\_layers, by.x = id, by.y = id) %>%   
 select(observacao\_id, coord\_x, coord\_y, taxon\_sibcs\_2009, argila\_)  
rm(dnos\_layers, dnos\_observations)  
sp::coordinates(dnos) <- ~ coord\_x + coord\_y  
sp::proj4string(dnos) <- sirgas2000  
dnos <- sp::spTransform(dnos, wgs84utm22s)  
  
dnos$um <- sp::over(x = dnos, y = pedologia25) %>% unlist()  
  
dnos\_in <- dnos[!is.na(dnos$um), ]

Será plotado o mapa com a localização dos pontos amostrais distribuidos sobre o mapa pedológico da área. Para isso será usada a função . Notemos que, anteriormente, já havia sido criada uma rampa de cores específica para pedologia25, constituída de tantas cores quantas foram as unidades de mapeamento.

sp::spplot(  
 pedologia25, col.regions = col\_soil\_um, alpha.regions = 0.3, colorkey = FALSE) +  
 latticeExtra::as.layer(  
 lattice::xyplot(coord\_y ~ coord\_x, data = as.data.frame(dnos\_in@coords), col = 'blue', pch = 16))



Pode ser verificada a distribuição dos pontos amostrais com maior densidade em algumas regiões. Isso possivelmente é decorrência da facilidade de acesso, seja pela presença de vias ou pelas caracteristicas do terreno, que por sua vez podem dificultar ou impossibilitar o acesso.

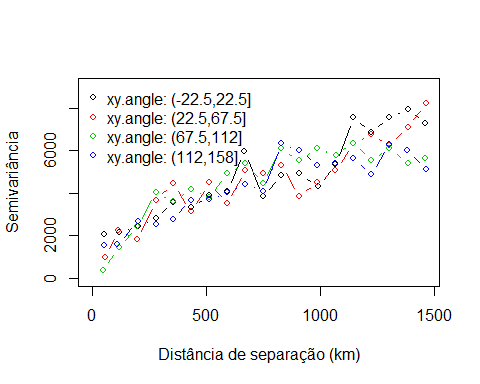
# Modelagem puramente espacial

Nesta etapa será realizada a predição do conteúdo de argila no solo na bacia do DNOS usando krigagem ordinária. Essa prática consiste em pressupor que o valor médio de uma variável é constante em curtos intervalos de distância, fazendo a predição espacial tanto dos efeitos aleatórios como da média espacial. Primeiramente, será calculada uma regressão linear do valor de argila em função apenas de seu próprio valor.

ko <- lm(argila\_ ~ 1, dnos\_in)  
summary(ko)

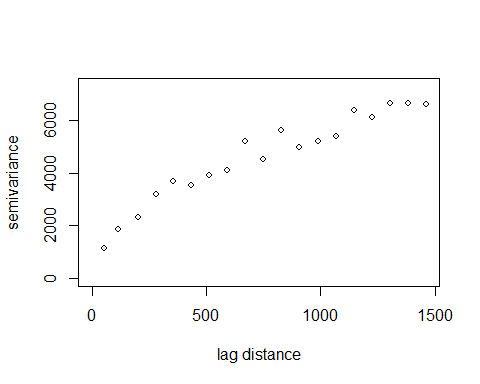
##   
## Call:  
## lm(formula = argila\_ ~ 1, data = dnos\_in)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -166.24 -90.49 -28.24 78.76 587.76   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 176.245 5.662 31.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 109.8 on 375 degrees of freedom

limites <- seq(0, 1500, length.out = 20)  
residuals(ko) %>%   
 georob::sample.variogram(  
 locations = dnos\_in@coords, lag.dist.def = limites,  
 xy.angle.def = c(0, 22.5, 67.5, 112.5, 157.5, 180)) %>%   
 plot(type = "b", ylab = 'Semivariância', xlab = 'Distância de separação (km)')



Conforme mostra a figura anterior, os resíduos da regressão linear possuem dependência espacial. Partindo da primeira classe de distância de separação, a semivariância aumenta de maneira relativamente rápida. Por não haver existência de estruturas de autocorrelação espacial dependentes da direção, foi assumida a isotropia do processo, permitindo assim criar o semivariograma independente de direção.

vario\_ko <-   
 residuals(ko) %>%   
 georob::sample.variogram(  
 locations = dnos\_in@coords, lag.dist.def = limites)  
plot(vario\_ko)



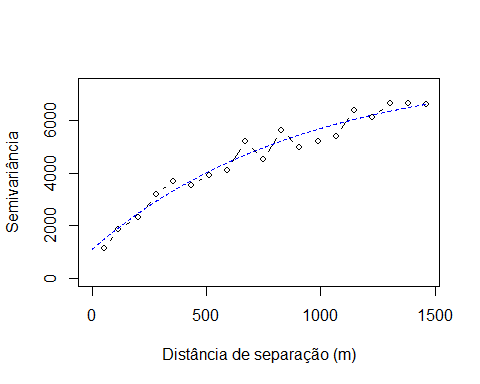
Em seguida, será ajustado ao variograma amostral um modelo exponencial do variograma. Será usado o método dos quadrados mínimos não-lineares ponderados, conforme o método de . A estimativa dos parâmetros do variograma é conduzido por otimização, utilizando a função $stats::optim(method = "BFGS"$). A partir da visualização do variograma experimental, precisamos estimar a , o e a .

vario\_fit\_ko <-   
 georob::fit.variogram.model(  
 vario\_ko, variogram.model = 'RMexp', param = c(variance = 5500, nugget = 500, scale = 200),   
 weighting.method = "cressie", method = "BFGS")

##   
##   
## variance snugget nugget scale  
## Variogram parameters 0.000000e+00 0.000000e+00 4.611987e-99 1.759151e+303  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.279835e-174 0.000000e+00 3.104228e-18 3.089474e+62  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.635765e-32 0.000000e+00 4.545344e-02 2.181730e+14  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 4.105503e+19 0.000000e+00 3.562234e-12 6.651118e+274  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 7.404391e+06 0.000000e+00 7.317243e-01 6.619288e+56  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 2.092733e+04 0.000000e+00 1.339283e+02 1.661095e+13  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 4.536952e+42 0.000000e+00 9.778529e+05 8.113002e+02  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 5.923150e+08 0.000000e+00 1.302407e+06 7.093282e+16  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 6.463195e+02 0.000000e+00 4.204099e+07 1.270991e+12  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.935884e+22 0.000000e+00 2.054224e+35 1.656279e+46  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 7.112791e+03 0.000000e+00 1.238562e+12 2.009767e+03  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 2.545136e+05 0.000000e+00 Inf 5.587137e+33  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.376638e+04 0.000000e+00 3.250345e+104 8.498906e+08  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 7.681419e+03 0.000000e+00 5.980718e+22 9.242677e+03  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 6.781484e-240 0.000000e+00 6.212981e-12 1.743945e+141  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.632008e-45 0.000000e+00 6.457879e-01 2.657465e+30  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.985501e+18 0.000000e+00 7.036068e+09 1.547975e+41  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 6.949823e+03 0.000000e+00 1.489679e+19 1.845578e+06  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 2.662572e-104 0.000000e+00 2.035860e-37 1.219345e+73  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 2.241812e-18 0.000000e+00 1.274794e-05 9.395050e+16

É possível verificar a concordância do modelo escolhido, onde a curva se assemelha com a caracteristica dos dados.

plot(vario\_ko, type = "b", xlab = 'Distância de separação (m)', ylab = 'Semivariância')  
lines(vario\_fit\_ko, col = "blue", lty = 'dashed')



Com o semivariograma estimado, agora será realizado o ajuste.

reml\_fit\_ko <- georob::georob(  
 argila\_ ~ 1, dnos\_in, locations = ~ coord\_x + coord\_y,   
 variogram.model = 'RMexp',   
 param = c(variance = vario\_fit\_ko$variogram.object[[1]]$param[['variance']],   
 nugget = vario\_fit\_ko$variogram.object[[1]]$param[['nugget']],   
 scale = vario\_fit\_ko$variogram.object[[1]]$param[['scale']]),  
 tuning.psi = 1000, control = georob::control.georob(initial.fixef = 'lm'))  
summary(reml\_fit\_ko)

##   
## Call:georob::georob(formula = argila\_ ~ 1, data = dnos\_in, locations = ~coord\_x +   
## coord\_y, variogram.model = "RMexp", param = c(variance = vario\_fit\_ko$variogram.object[[1]]$param[["variance"]],   
## nugget = vario\_fit\_ko$variogram.object[[1]]$param[["nugget"]],   
## scale = vario\_fit\_ko$variogram.object[[1]]$param[["scale"]]),   
## tuning.psi = 1000, control = georob::control.georob(initial.fixef = "lm"))  
##   
## Tuning constant: 1000   
##   
## Convergence in 7 function and 6 Jacobian/gradient evaluations  
##   
## Estimating equations (gradient)  
##   
## eta scale  
## Gradient : 1.284280e-03 1.828333e-03  
##   
## Maximized restricted log-likelihood: -2144.942   
##   
## Predicted latent variable (B):  
## Min 1Q Median 3Q Max   
## -188.83 -141.25 -82.98 13.69 412.66   
##   
## Residuals (epsilon):  
## Min 1Q Median 3Q Max   
## -57.8646 -11.6304 -0.6168 9.9875 117.7777   
##   
## Standardized residuals:  
## Min 1Q Median 3Q Max   
## -2.84757 -0.62857 -0.03244 0.55007 7.40387   
##   
##   
## Gaussian REML estimates  
##   
## Variogram: RMexp   
## Estimate Lower Upper  
## variance 13765.8 7613.9 24888  
## snugget(fixed) 0.0 NA NA  
## nugget 1159.0 479.3 2802  
## scale 740.1 324.4 1689  
##   
##   
## Fixed effects coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 233.56 34.98 6.678 8.76e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error (sqrt(nugget)): 34.04   
##   
## Robustness weights:   
## All 376 weights are ~= 1.

grid <- sp::spsample(dnos\_in, 10000, type = 'regular')  
grid <-   
 sp::SpatialPointsDataFrame(  
 coords = grid@coords,   
 data = data.frame(  
 um = sp::over(grid, pedologia25) %>% unlist()),  
 proj4string = grid@proj4string)

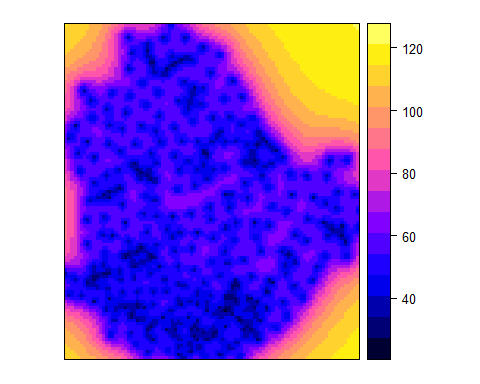
Estimado o semivariograma, será realizada a predição espacial do valor de argila, com base nos parâmetros fixados no semivariogrâma. A predição será realizada pelo método , com significância de 95%.

colnames(grid@coords) <- colnames(dnos\_in@coords)  
pred\_ponto\_ko <- predict(  
 reml\_fit\_ko, newdata = grid, type = 'signal', signif = 0.95,  
 control = georob::control.predict.georob(extended.output = TRUE))  
sp::gridded(pred\_ponto\_ko) <- TRUE  
str(pred\_ponto\_ko)

## Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots  
## ..@ data :'data.frame': 10058 obs. of 8 variables:  
## .. ..$ pred : num [1:10058] 214 214 213 213 212 ...  
## .. ..$ se : num [1:10058] 112 111 110 109 108 ...  
## .. ..$ lower : num [1:10058] -4.4 -3.453 -2.316 -0.962 0.634 ...  
## .. ..$ upper : num [1:10058] 433 431 429 426 424 ...  
## .. ..$ trend : num [1:10058] 234 234 234 234 234 ...  
## .. ..$ var.pred : num [1:10058] 2685 2824 2976 3144 3330 ...  
## .. ..$ cov.pred.target: num [1:10058] 1989 2149 2324 2515 2724 ...  
## .. ..$ var.target : num [1:10058] 13766 13766 13766 13766 13766 ...  
## .. ..- attr(\*, "variogram.object")=List of 1  
## .. .. ..$ :List of 9  
## .. .. .. ..$ variogram.model: chr "RMexp"  
## .. .. .. ..$ param : Named num [1:4] 13766 0 1159 740  
## .. .. .. .. ..- attr(\*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"  
## .. .. .. ..$ fit.param : Named logi [1:4] TRUE FALSE TRUE TRUE  
## .. .. .. .. ..- attr(\*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"  
## .. .. .. ..$ isotropic : logi TRUE  
## .. .. .. ..$ aniso : Named num [1:5] 1 1 90 90 0  
## .. .. .. .. ..- attr(\*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...  
## .. .. .. ..$ fit.aniso : Named logi [1:5] FALSE FALSE FALSE FALSE FALSE  
## .. .. .. .. ..- attr(\*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...  
## .. .. .. ..$ sincos :List of 6  
## .. .. .. .. ..$ co: num 6.12e-17  
## .. .. .. .. ..$ so: num 1  
## .. .. .. .. ..$ cp: num 6.12e-17  
## .. .. .. .. ..$ sp: num 1  
## .. .. .. .. ..$ cz: num 1  
## .. .. .. .. ..$ sz: num 0  
## .. .. .. ..$ rotmat : num [1:2, 1:2] 1.00 -6.12e-17 6.12e-17 1.00  
## .. .. .. ..$ sclmat : Named num [1:2] 1 1  
## .. .. .. .. ..- attr(\*, "names")= chr [1:2] "" "f1"  
## .. ..- attr(\*, "psi.func")= chr "logistic"  
## .. ..- attr(\*, "tuning.psi")= num 1000  
## .. ..- attr(\*, "type")= chr "signal"  
## ..@ coords.nrs : num(0)   
## ..@ grid :Formal class 'GridTopology' [package "sp"] with 3 slots  
## .. .. ..@ cellcentre.offset: Named num [1:2] 226855 6715906  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..@ cellsize : Named num [1:2] 52.7 52.7  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..@ cells.dim : Named int [1:2] 94 107  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## ..@ grid.index : int [1:10058] 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 ...  
## ..@ coords : num [1:10058, 1:2] 226855 226908 226960 227013 227066 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:10058] "um1" "um2" "um3" "um4" ...  
## .. .. ..$ : chr [1:2] "coord\_x" "coord\_y"  
## ..@ bbox : num [1:2, 1:2] 226828 6715880 231782 6721519  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..$ : chr [1:2] "min" "max"  
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot  
## .. .. ..@ projargs: chr NA

A figura abaixo demonstra os valores de argila preditos e espacializados por krigagem na área de estudo.

krig\_ko <- sp::spplot(pred\_ponto\_ko, zcol = 'se')  
plot(krig\_ko)



Contudo, para que tenhamos uma medida de acurácia dos dados, será aplicada a validação cruzada aos dados gerados. O procedimento da validação cruzada consiste na partição aleatória do conjunto completo de dados em subconjuntos. A cada passo da validação cruzada, um dos subconjuntos é utilizado apenas como conjunto de dados de validação. Os demais subconjuntos são utilizados para constituir o conjunto de calibração. Com o modelo calibrado, faz-se a predição dos valores de argila nas observações do conjunto de dados de validação. Esse procedimento é repetido até que cada subconjunto seja, em algum momento, deixado separado para constituir o conjunto de dados de validação enquanto os outros conjuntos são utilizados para calibração do modelo. Com as predições realizadas para cada um dos subconjuntos, são calculados os erros para avaliar a qualidade das predições.

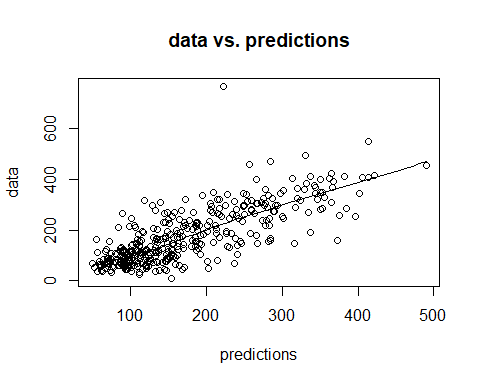
validacao\_ko <- georob::cv(reml\_fit\_ko, nset = 375)  
summary(validacao\_ko)

##   
## Statistics of cross-validation prediction errors  
## me mede rmse made qne msse medsse crps   
## 1.2476 -1.5714 70.0936 56.5017 59.8017 1.1272 0.3558 36.5633

1 - sum((validacao\_ko$pred$data - validacao\_ko$pred$pred)^2) /   
 sum((validacao\_ko$pred$data - mean(validacao\_ko$pred$data))^2)

## [1] 0.5912629

plot(validacao\_ko)



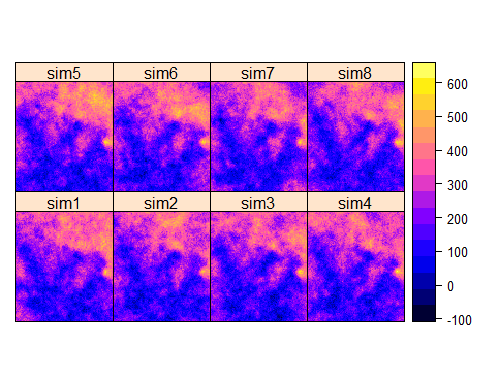
As estatisticas de erro geradas pela validação cruzada serão utilizadas para comparação com os demais modelos, permitindo avaliar a eficiência de ambos.

Serão gerados mapas contendo a incerteza da predição realizada. O grau de incerteza contido em um mapa é fundamental para avaliar a qualidade da informação contida no mapa produzido e, conseqüentemente, verificar o grau de adequação para usos posteriores.

m\_exp\_ko <- coef(reml\_fit\_ko, what = 'variogram')  
m\_exp\_ko <- vgm(psill = 5900, model = 'Exp', range = 1200, nugget = 500)  
cond\_sim\_ko <- gstat::krige(  
 argila\_ ~ 1, locations = dnos\_in, newdata = grid, model = m\_exp\_ko, nmax = 15, nsim = 8)

## drawing 8 GLS realisations of beta...  
## [using conditional Gaussian simulation]

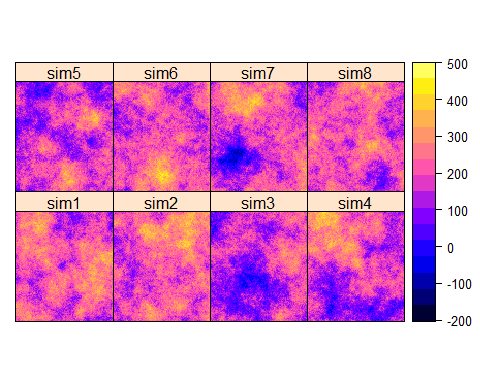
gridded(cond\_sim\_ko) <- TRUE  
spplot(cond\_sim\_ko)



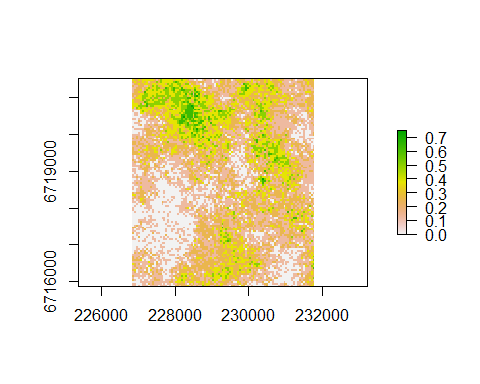
beta <- mean(dnos\_in$argila\_)  
m\_exp\_ko <- coef(reml\_fit\_ko, what = 'variogram')  
m\_exp\_ko <- vgm(psill = 5900, model = 'Exp', range = 1200, nugget = 500)  
g <- gstat::gstat(formula = z ~ 1, dummy = TRUE, model = m\_exp\_ko, nmax = 15, beta = beta)  
uncond\_sim\_ko <- predict(g, newdata = grid, nsim = 8)

## [using unconditional Gaussian simulation]

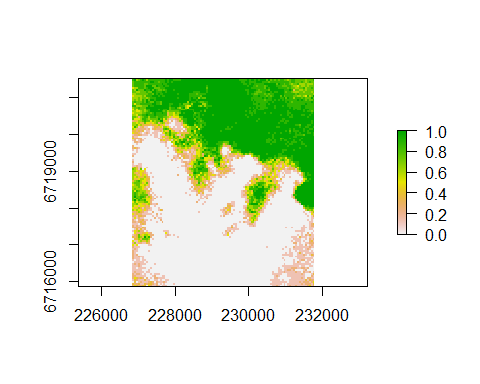
gridded(uncond\_sim\_ko) <- TRUE  
spplot(uncond\_sim\_ko)



uncond\_sim\_prop\_ko <-  
 raster::calc(stack(uncond\_sim\_ko), function (x) x > 250) %>%  
 calc(sum) %>%  
calc(function(x) x / 8)  
plot(uncond\_sim\_prop\_ko, asp = 1)



cond\_sim\_prop\_ko <-  
 raster::calc(stack(cond\_sim\_ko), function (x) x > 250) %>%  
 calc(sum) %>%  
calc(function(x) x / 8)  
plot(cond\_sim\_prop\_ko, asp = 1)



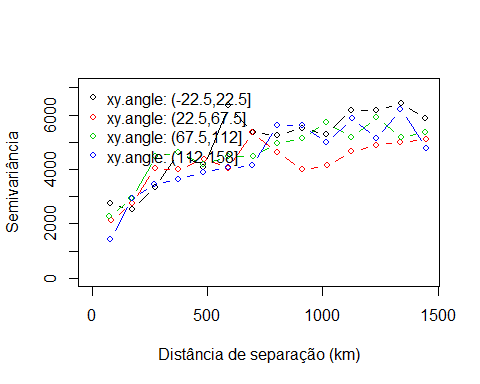
# Modelagem espacial, fazendo uso de krigagem universal utilizando o mapa pedológico como covariável.

Nesse caso, assumiremos que o processo estocástico não é estacionário, ou seja, o processo apresenta uma tendência, nesse caso representado pela predição dos valores de argila em função das unidades de mapeamento do mapa pedológico como preditoras, inserido pela função . O processo de modelagem é o mesmo realizado anteriormente na krigagem ordinária, alterando apenas o ajuste do semivariograma pelo efeito das informações contidas no mapa pedológico.

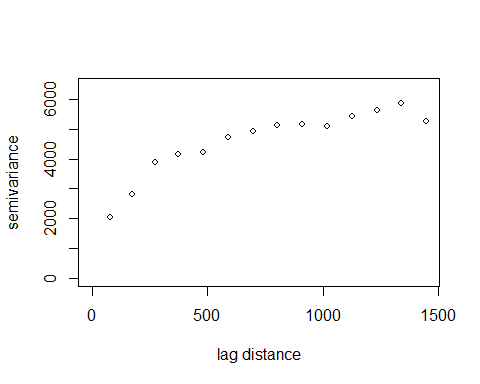
ku <- lm(argila\_ ~ um, dnos\_in)  
summary(ku)

##   
## Call:  
## lm(formula = argila\_ ~ um, data = dnos\_in)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -209.82 -60.57 -4.50 50.32 506.29   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 101.412 9.107 11.136  
## umArgissolo Vermelho 240.403 16.048 14.980  
## umCambissolo - Neossolo 156.302 14.777 10.578  
## umNeossolo Flúvico -38.413 58.314 -0.659  
## umNeossolo Litólico 47.156 11.054 4.266  
## umNeossolo Litólico - Neossolo Regolí­tico 140.587 24.358 5.772  
## umNeossolo Regolí­tico -39.163 41.734 -0.938  
## umNeossolo Regolítico 66.587 30.205 2.205  
## umPlanossolo Háplico 95.895 24.358 3.937  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## umArgissolo Vermelho < 2e-16 \*\*\*  
## umCambissolo - Neossolo < 2e-16 \*\*\*  
## umNeossolo Flúvico 0.5105   
## umNeossolo Litólico 2.54e-05 \*\*\*  
## umNeossolo Litólico - Neossolo Regolí­tico 1.67e-08 \*\*\*  
## umNeossolo Regolí­tico 0.3487   
## umNeossolo Regolítico 0.0281 \*   
## umPlanossolo Háplico 9.88e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 81.46 on 367 degrees of freedom  
## Multiple R-squared: 0.4612, Adjusted R-squared: 0.4495   
## F-statistic: 39.27 on 8 and 367 DF, p-value: < 2.2e-16

limites <- seq(0, 1500, length.out = 15)  
residuals(ku) %>%   
 georob::sample.variogram(  
 locations = dnos\_in@coords, lag.dist.def = limites,  
 xy.angle.def = c(0, 22.5, 67.5, 112.5, 157.5, 180)) %>%   
 plot(type = "b", ylab = 'Semivariância', xlab = 'Distância de separação (km)')



vario\_ku <-   
 residuals(ku) %>%   
 georob::sample.variogram(  
 locations = dnos\_in@coords, lag.dist.def = limites)  
plot(vario\_ku)



vario\_fit\_ku <-   
 georob::fit.variogram.model(  
 vario\_ku, variogram.model = 'RMexp', param = c(variance = 4000, nugget = 1000, scale = 200),   
 weighting.method = "cressie", method = "BFGS")

##   
##   
## variance snugget nugget scale  
## Variogram parameters Inf 0.000000e+00 4.374752e+148 2.260996e+49  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.321032e+131 0.000000e+00 1.343355e+32 5.148538e+11  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.275941e+29 0.000000e+00 6.693267e+08 1.524617e+04  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 1.078907e+15 0.000000e+00 2.233763e-10 7.145381e+102  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 8.334694e+05 0.000000e+00 3.000988e+00 2.594408e+22  
##   
##   
## variance snugget nugget scale  
## Variogram parameters 5.245660e-01 0.000000e+00 5.335079e+20 1.040535e+09

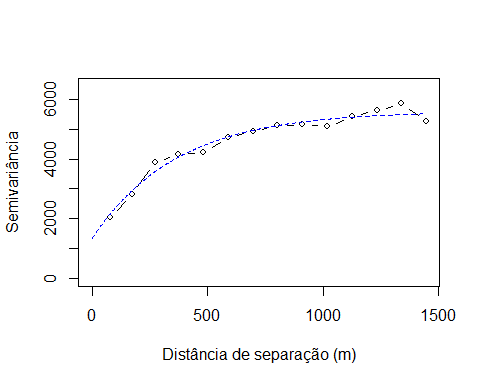
summary(vario\_fit\_ku)

##   
## Call:georob::fit.variogram.model(sv = vario\_ku, variogram.model = "RMexp",   
## param = c(variance = 4000, nugget = 1000, scale = 200), weighting.method = "cressie",   
## method = "BFGS")  
##   
## Convergence in 61 function and 13 Jacobian/gradient evaluations  
##   
## Residual Sum of Squares: 31.80945   
##   
## Residuals (epsilon):  
## Min 1Q Median 3Q Max   
## -255.73 -150.52 -38.10 70.76 361.74   
##   
## Variogram: RMexp   
## Estimate Lower Upper  
## variance 4259.2 3951.1 4591.3  
## snugget(fixed) 0.0 NA NA  
## nugget 1343.6 1013.2 1781.8  
## scale 367.7 314.3 430.1

reml\_ku <- georob::georob(  
 argila\_ ~ um, dnos\_in, locations = ~ coord\_x + coord\_y,   
 variogram.model = 'RMexp',   
 param = c(variance = vario\_fit\_ku$variogram.object[[1]]$param[['variance']],   
 nugget = vario\_fit\_ku$variogram.object[[1]]$param[['nugget']],   
 scale = vario\_fit\_ku$variogram.object[[1]]$param[['scale']]),  
 tuning.psi = 1000, control = georob::control.georob(initial.fixef = 'lm'))  
summary(reml\_ku)

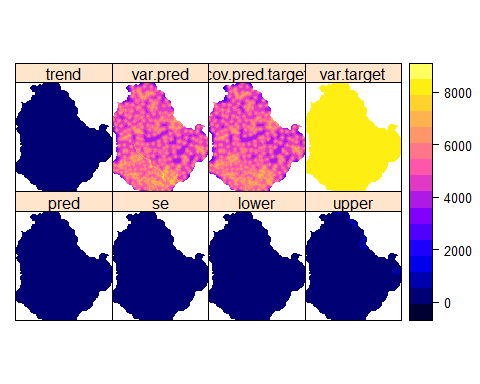
##   
## Call:georob::georob(formula = argila\_ ~ um, data = dnos\_in, locations = ~coord\_x +   
## coord\_y, variogram.model = "RMexp", param = c(variance = vario\_fit\_ku$variogram.object[[1]]$param[["variance"]],   
## nugget = vario\_fit\_ku$variogram.object[[1]]$param[["nugget"]],   
## scale = vario\_fit\_ku$variogram.object[[1]]$param[["scale"]]),   
## tuning.psi = 1000, control = georob::control.georob(initial.fixef = "lm"))  
##   
## Tuning constant: 1000   
##   
## Convergence in 7 function and 7 Jacobian/gradient evaluations  
##   
## Estimating equations (gradient)  
##   
## eta scale  
## Gradient : -3.899355e-04 4.143282e-02  
##   
## Maximized restricted log-likelihood: -2101.01   
##   
## Predicted latent variable (B):  
## Min 1Q Median 3Q Max   
## -142.99 -90.02 -44.37 33.90 438.75   
##   
## Residuals (epsilon):  
## Min 1Q Median 3Q Max   
## -47.026 -9.706 -0.139 8.633 100.992   
##   
## Standardized residuals:  
## Min 1Q Median 3Q Max   
## -2.547501 -0.617061 -0.009435 0.514866 7.294817   
##   
##   
## Gaussian REML estimates  
##   
## Variogram: RMexp   
## Estimate Lower Upper  
## variance 7975.1 5464.4 11639.2  
## snugget(fixed) 0.0 NA NA  
## nugget 1032.5 350.3 3043.5  
## scale 370.8 199.2 690.2  
##   
##   
## Fixed effects coefficients:  
## Estimate Std. Error t value  
## (Intercept) 172.49 21.58 7.994  
## umArgissolo Vermelho 112.84 24.70 4.568  
## umCambissolo - Neossolo 51.77 21.84 2.370  
## umNeossolo Flúvico -44.95 39.63 -1.134  
## umNeossolo Litólico 12.99 16.14 0.805  
## umNeossolo Litólico - Neossolo Regolí­tico 63.15 34.90 1.809  
## umNeossolo Regolí­tico -28.58 37.73 -0.758  
## umNeossolo Regolítico 46.01 32.63 1.410  
## umPlanossolo Háplico 42.27 24.83 1.702  
## Pr(>|t|)   
## (Intercept) 1.71e-14 \*\*\*  
## umArgissolo Vermelho 6.73e-06 \*\*\*  
## umCambissolo - Neossolo 0.0183 \*   
## umNeossolo Flúvico 0.2574   
## umNeossolo Litólico 0.4216   
## umNeossolo Litólico - Neossolo Regolí­tico 0.0712 .   
## umNeossolo Regolí­tico 0.4492   
## umNeossolo Regolítico 0.1594   
## umPlanossolo Háplico 0.0896 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error (sqrt(nugget)): 32.13   
##   
## Robustness weights:   
## All 376 weights are ~= 1.

plot(vario\_ku, type = "b", xlab = 'Distância de separação (m)', ylab = 'Semivariância')  
lines(vario\_fit\_ku, col = "blue", lty = 'dashed')



grid <- sp::spsample(dnos\_in, 10000, type = 'regular')  
grid <-   
 sp::SpatialPointsDataFrame(  
 coords = grid@coords,   
 data = data.frame(  
 um = sp::over(grid, pedologia25) %>% unlist()),  
 proj4string = grid@proj4string)

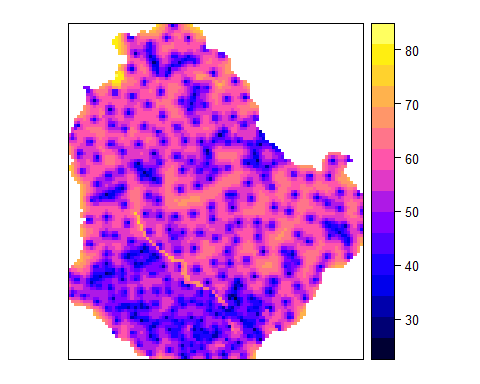
colnames(grid@coords) <- colnames(dnos\_in@coords)  
pred\_ku <- predict(  
 reml\_ku, newdata = grid, type = 'signal', signif = 0.95,  
 control = georob::control.predict.georob(extended.output = TRUE))  
sp::gridded(pred\_ku) <- TRUE  
spplot(pred\_ku)



str(pred\_ku)

## Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots  
## ..@ data :'data.frame': 10058 obs. of 8 variables:  
## .. ..$ pred : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ se : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ lower : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ upper : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ trend : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ var.pred : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ cov.pred.target: num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..$ var.target : num [1:10058] NA NA NA NA NA NA NA NA NA NA ...  
## .. ..- attr(\*, "variogram.object")=List of 1  
## .. .. ..$ :List of 9  
## .. .. .. ..$ variogram.model: chr "RMexp"  
## .. .. .. ..$ param : Named num [1:4] 7975 0 1033 371  
## .. .. .. .. ..- attr(\*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"  
## .. .. .. ..$ fit.param : Named logi [1:4] TRUE FALSE TRUE TRUE  
## .. .. .. .. ..- attr(\*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"  
## .. .. .. ..$ isotropic : logi TRUE  
## .. .. .. ..$ aniso : Named num [1:5] 1 1 90 90 0  
## .. .. .. .. ..- attr(\*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...  
## .. .. .. ..$ fit.aniso : Named logi [1:5] FALSE FALSE FALSE FALSE FALSE  
## .. .. .. .. ..- attr(\*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...  
## .. .. .. ..$ sincos :List of 6  
## .. .. .. .. ..$ co: num 6.12e-17  
## .. .. .. .. ..$ so: num 1  
## .. .. .. .. ..$ cp: num 6.12e-17  
## .. .. .. .. ..$ sp: num 1  
## .. .. .. .. ..$ cz: num 1  
## .. .. .. .. ..$ sz: num 0  
## .. .. .. ..$ rotmat : num [1:2, 1:2] 1.00 -6.12e-17 6.12e-17 1.00  
## .. .. .. ..$ sclmat : Named num [1:2] 1 1  
## .. .. .. .. ..- attr(\*, "names")= chr [1:2] "" "f1"  
## .. ..- attr(\*, "psi.func")= chr "logistic"  
## .. ..- attr(\*, "tuning.psi")= num 1000  
## .. ..- attr(\*, "type")= chr "signal"  
## ..@ coords.nrs : num(0)   
## ..@ grid :Formal class 'GridTopology' [package "sp"] with 3 slots  
## .. .. ..@ cellcentre.offset: Named num [1:2] 226870 6715906  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..@ cellsize : Named num [1:2] 52.7 52.7  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..@ cells.dim : Named int [1:2] 94 107  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "coord\_x" "coord\_y"  
## ..@ grid.index : int [1:10058] 9965 9966 9967 9968 9969 9970 9971 9972 9973 9974 ...  
## ..@ coords : num [1:10058, 1:2] 226870 226923 226975 227028 227081 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:10058] "um1" "um2" "um3" "um4" ...  
## .. .. ..$ : chr [1:2] "coord\_x" "coord\_y"  
## ..@ bbox : num [1:2, 1:2] 226844 6715880 231797 6721519  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:2] "coord\_x" "coord\_y"  
## .. .. ..$ : chr [1:2] "min" "max"  
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot  
## .. .. ..@ projargs: chr NA

krig\_ku <- sp::spplot(pred\_ku, zcol = 'se')  
plot(krig\_ku)



Validação cruzada do modelo utilizando apenas krigagem ordinária.

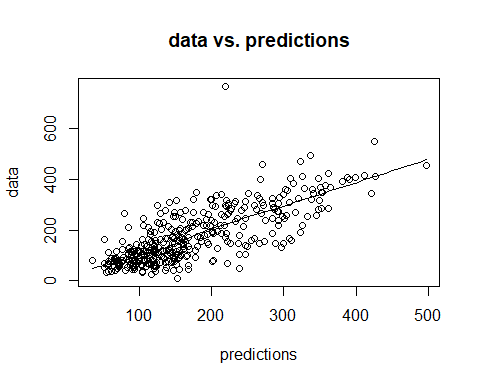
validacao\_ku <- georob::cv(reml\_ku, nset = 375)  
summary(validacao\_ku)

##   
## Statistics of cross-validation prediction errors  
## me mede rmse made qne msse medsse crps   
## 1.2843 -0.6265 70.2853 56.9705 61.0215 1.1174 0.3473 36.9357

1 - sum((validacao\_ku$pred$data - validacao\_ku$pred$pred)^2) /   
 sum((validacao\_ku$pred$data - mean(validacao\_ku$pred$data))^2)

## [1] 0.5890244

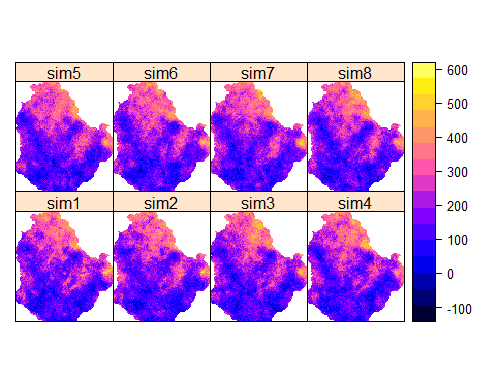
plot(validacao\_ku)



m\_exp\_ku <- coef(vario\_fit\_ku, what = 'variogram')  
m\_exp\_ku <- vgm(psill = 5000, model = 'Exp', range = 1200, nugget = 1000)  
cond\_sim\_ku <- gstat::krige(  
 argila\_ ~ um, locations = dnos\_in, newdata = grid, model = m\_exp\_ku, nmax = 15, nsim = 8)

## drawing 8 GLS realisations of beta...  
## [using conditional Gaussian simulation]

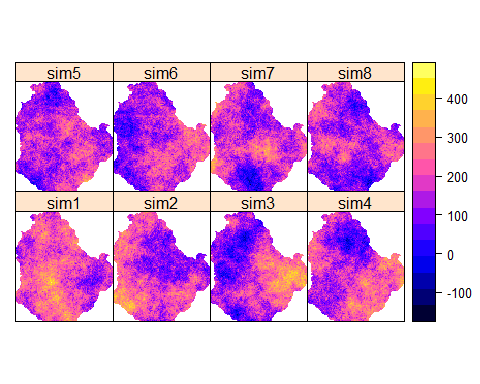
gridded(cond\_sim\_ku) <- TRUE  
spplot(cond\_sim\_ku)



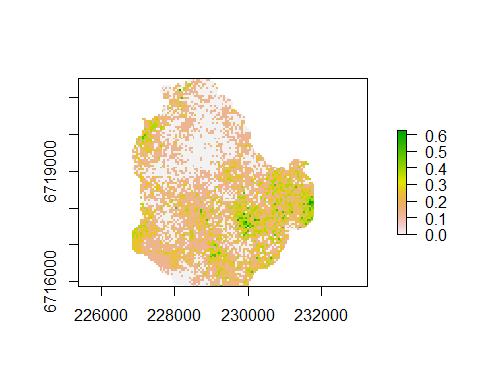
beta <- mean(dnos\_in$argila\_)  
m\_exp\_ku <- coef(vario\_fit\_ko, what = 'variogram')  
m\_exp\_ku <- vgm(psill = 5000, model = 'Exp', range = 1200, nugget = 1000)  
g\_ku <- gstat::gstat(formula = z ~ um, dummy = TRUE, model = m\_exp\_ku, nmax = 15, beta = beta)  
uncond\_sim\_ku <- predict(g\_ku, newdata = grid, nsim = 8)

## [using unconditional Gaussian simulation]

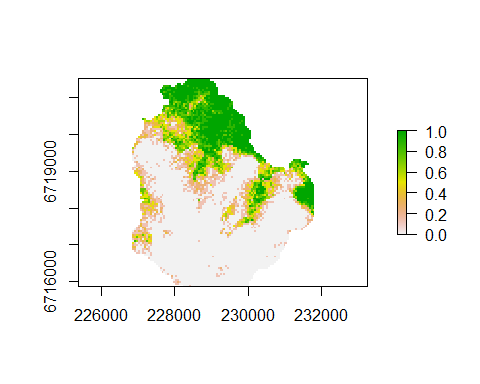
gridded(uncond\_sim\_ku) <- TRUE  
spplot(uncond\_sim\_ku)



uncond\_sim\_prop\_ku <-  
 raster::calc(stack(uncond\_sim\_ku), function (x) x > 250) %>%  
 calc(sum) %>%  
calc(function(x) x / 8)  
plot(uncond\_sim\_prop\_ku, asp = 1)

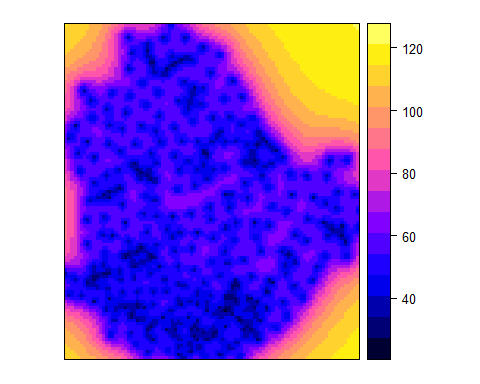


cond\_sim\_prop\_ku <-  
 raster::calc(stack(cond\_sim\_ku), function (x) x > 250) %>%  
 calc(sum) %>%  
calc(function(x) x / 8)  
plot(cond\_sim\_prop\_ku, asp = 1)

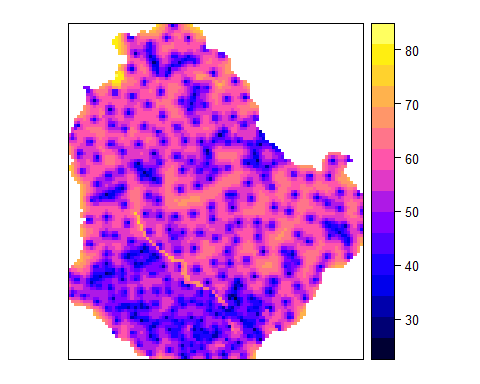


# Avaliação dos diferentes modelos na predição dos valores de argila

plot(krig\_ko, main = 'Valores de argila no solo via krigagem ordinaria')

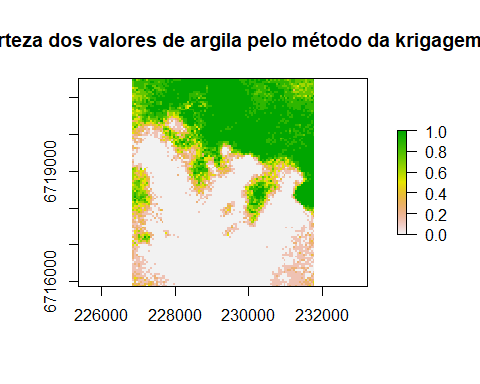


plot(krig\_ku, main = ' Valores de argila no solo via krigagem universal')

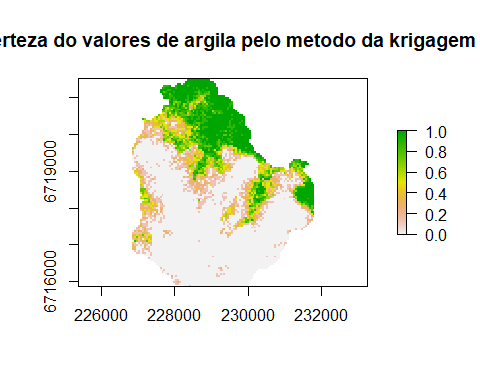


Apesar do mapa gerado via krigagem ordinária não apresentar o contorno da área e, devido a isso, ser mostrado em escala diferente, é possível verificar a semelhança entre ele e o mapa gerado via krigagem universal. Contudo, O mapa via krigagem universal demonstra a feição que se assemelha a de um rio, possivelmente advinda do mapa pedológico, que por sua vez serviu de covariável ambiental ao modelo linear usado para predição.

plot(cond\_sim\_prop\_ko, main = 'Mapa de incerteza dos valores de argila pelo método da krigagem ordinaria')

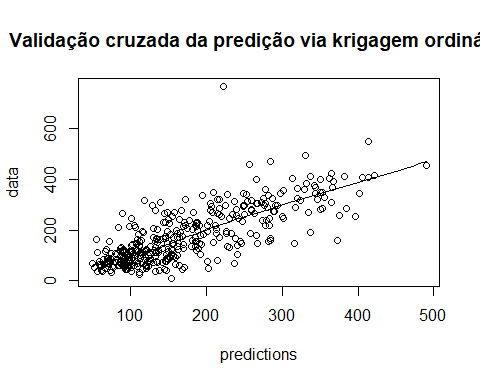


plot(cond\_sim\_prop\_ku, main = 'Mapa de incerteza do valores de argila pelo metodo da krigagem universal')

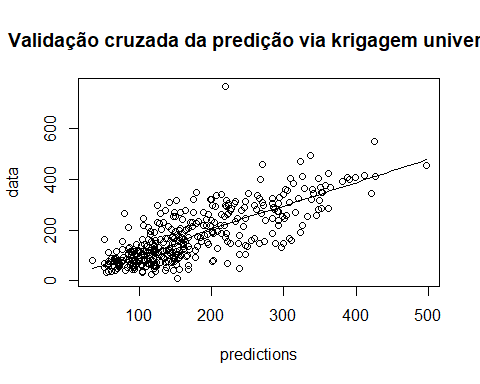


Apesar do mapa gerado pelo método da krigagem ordinária não possuir o contorno da área, podemos perceber que ambos os modelos apresentaram incertezas sem diferenças expressivas. Apesar de o modelo gerado via krigagem universal fazer uso de uma covariável ambiental, este se mostra com pontos de maior incerteza no centro do mapa.

plot(validacao\_ko, main = 'Validação cruzada da predição via krigagem ordinária')



plot(validacao\_ku, main = 'Validação cruzada da predição via krigagem universal')



valid <- c(summary(validacao\_ko, use.names = T), summary(validacao\_ku, use.names = T), recursive = T)  
valid

## st.me st.mede st.rmse st.made st.qne st.msse   
## 1.2475834 -1.5714125 70.0936459 56.5016895 59.8016563 1.1272412   
## st.medsse crps st.me st.mede st.rmse st.made   
## 0.3558026 36.5632584 1.2842553 -0.6264965 70.2853231 56.9704683   
## st.qne st.msse st.medsse crps   
## 61.0215241 1.1173873 0.3472833 36.9356690

A validação cruzada de ambos os modelos mostra-se com a linha de tendência passando em local próximo na nuvem de pontos. Como já demonstrado no mapa de incerteza, onde um número maior de pontos mostrou maior incerteza no modelo gerado via krigagem universal, as estatisticas de erro também demonstram um sensível aumento nos seus valores, como no da raiz do erro quadrático médio (), além de aumento no valor de pontuação de probabilidade nominal contínua (), que mede a distância entre as funções de densidade cumulativa prevista e observada.