Chrome_&_Zinc_spatial_analysis

May 25, 2017

0.1 Seminario :: Analisi statistica spaziale di alcune proprietà dei suoli

0.1.1 Laurea magistrale in scienze forestali ed ambientali

0.1.2 Corso Siti Contaminati

Giuliano Langella [glangella@unina.it]

0.1.3 Step #2: Esempio applicativo, analisi spaziale nella "Terra dei Fuochi" [Cromo & Zinco]

0.1.4 Premessa

Ordinary kriging (OK) The random field (RF) $\mathbf{Z}(\mathbf{u})$ is assumed to be intrinsic secondorder stationary if the first two moments (i.e. mean m and semivariance $\gamma(\mathbf{h})$) of the twopoint RF increments exist and are invariant under translation and rotation within a bounded area $\mathfrak{D}$ (Goovaerts1997_book, Wackernagel2003_book):

$$\begin{aligned} m &= E\{Z(\mathbf{u})\} \\ \gamma(\mathbf{h}) &= \frac{1}{2}E\left\{\left[Z\left(\mathbf{u}\right)-Z\left(\mathbf{u}+\mathbf{h}\right)\right]^2\right\} \end{aligned} \tag{1}$$

with theoretically infinite points locations $u(\mathbf{x}) \in \mathfrak{D}$, and random variables (RV) $Z(\mathbf{u})$ and $Z(\mathbf{u}+\mathbf{h})$ separated by the distance vector $h(\mathbf{x})$, where $\mathbf{x}$ represents the coordinates $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \in \Re^3$.

0.1.5 Inquadramento

0.1.6 D E F s

```
In [1]: # gdalwarp -tr 3000 3000 work/People/Michela/andosols_spatstat/dem_20m_ispra/dtm_20m_32.
        printf <- function(...) cat(sprintf(...))
```

In [34]:

```
# Gives count, mean, standard deviation, standard error of the mean, and confidence int
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {
    library(plyr)

    # New version of length which can handle NA's: if na.rm==T, don't count them
    length2 <- function (x, na.rm=FALSE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }

    # This does the summary. For each group's data frame, return a vector with
    # N, mean, and sd
    datac <- ddply(data, groupvars, .drop=.drop,
      .fun = function(xx, col) {
        c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
        )
      },
      measurevar
    )

    # Rename the "mean" column
    datac <- rename(datac, c("mean" = measurevar))

    datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult

    return(datac)
}
```

Requirements:

In [ ]:

```
#_________________
#
# P A C K A G E S
#_________________
require("ggplot2")
require("gridExtra")
require("dismo")
library("gstat")
require("rgdal")
```

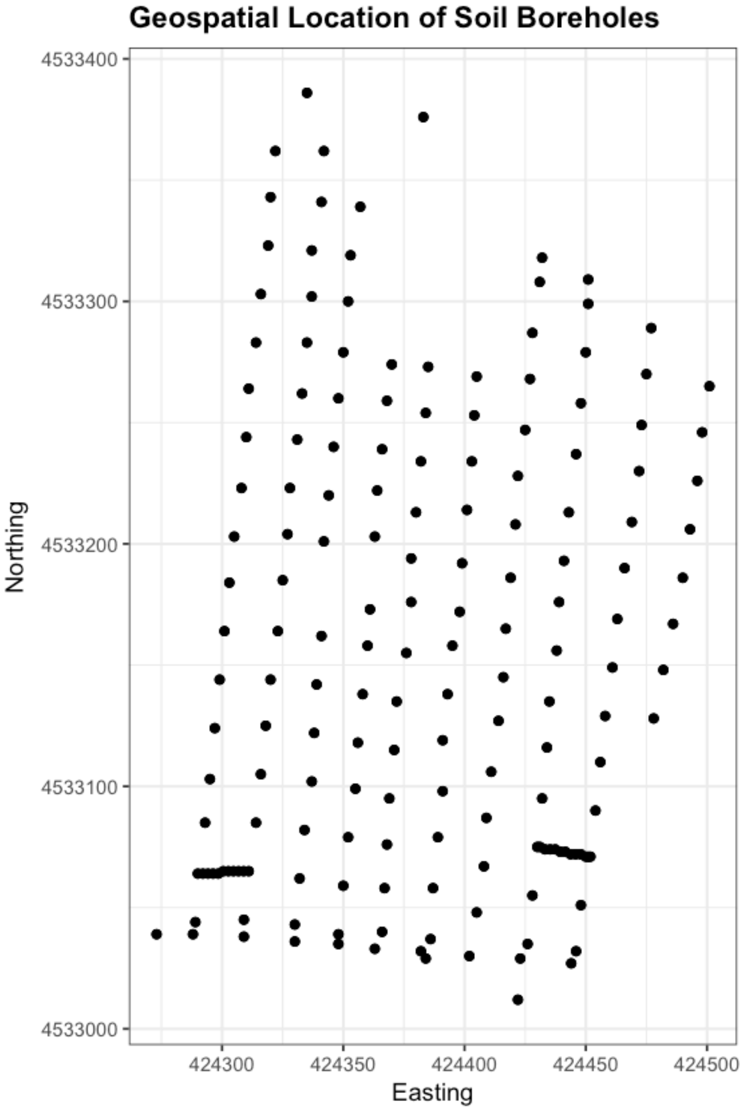
```
library("sp")
library("raster")
library("geoR")
```

Import data (San Giuseppiello Proj):

```
In [27]: # R E A D
         # xrf measurements
         chrome                 <- read.table("~/git/seminars/Siti_Contaminati/Chrome-xrf.txt",h
         zinc                   <- read.table("~/git/seminars/Siti_Contaminati/Zinc-xrf.txt",heade
         # grid for interpolations
         grd                    <- read.table("~/git/seminars/Siti_Contaminati/xrf_grid.txt",header
```

Geospatial location of soil pits (Biosoil Proj):

```
In [6]: # see
        #  http://stackoverflow.com/questions/41787313/how-to-create-a-grid-of-spatial-points
        #  https://pakillo.github.io/R-GIS-tutorial/
        it <- getData('GADM', country = 'IT', level = 1)
        campania <- it[it$NAME_1 == "Campania",]
        qplot(Easting, Northing, data=zinc) + coord_fixed(ratio = 1) + theme_bw() + ggtitle("Geo
```



Analisi statistica preliminare:

```
In [28]: names(chrome)
         names(zinc)
```

```
1. ’Easting’ 2. ’Northing’ 3. ’Layer’ 4. ’chrome’ 5. ’chromethre’ 6. ’std’
1. ’Easting’ 2. ’Northing’ 3. ’Layer’ 4. ’zinc’ 5. ’zincthre’ 6. ’std’
```

```
In [29]: summary(chrome[,c(4)])
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
37.0   138.0   333.0   468.7   583.5  3452.0
```

```
In [30]: summary(zinc[,c(4)])
```

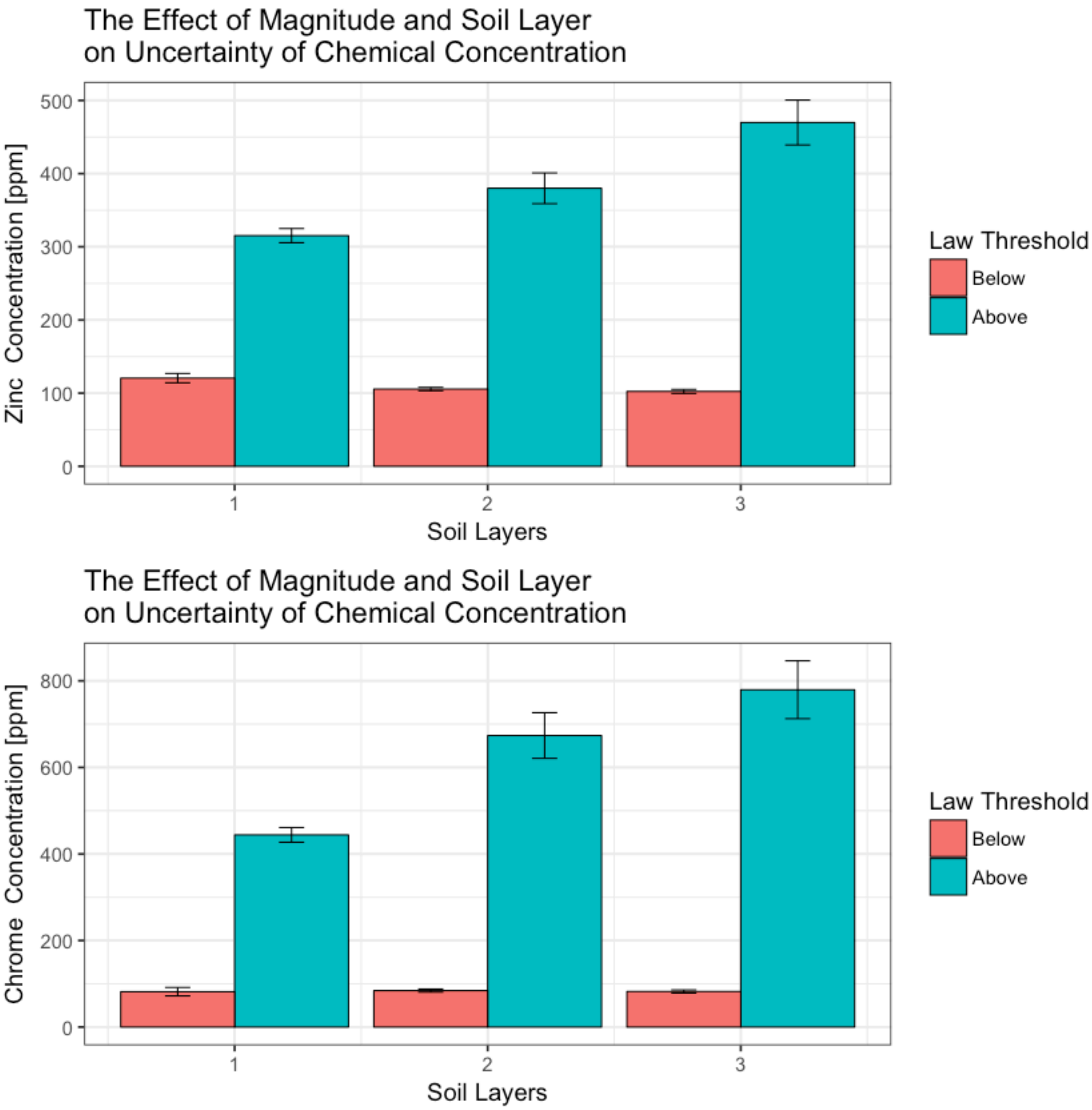
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   62.0   147.0   245.5   307.5   371.5  1490.0 
```

```
In [33]: #cor( D[,17],D[,c(1,11:16)] ) # , na.rm = TRUE
         cor( chrome$chrome,zinc$zinc ) # , na.rm = TRUE
```

0.899470615086919

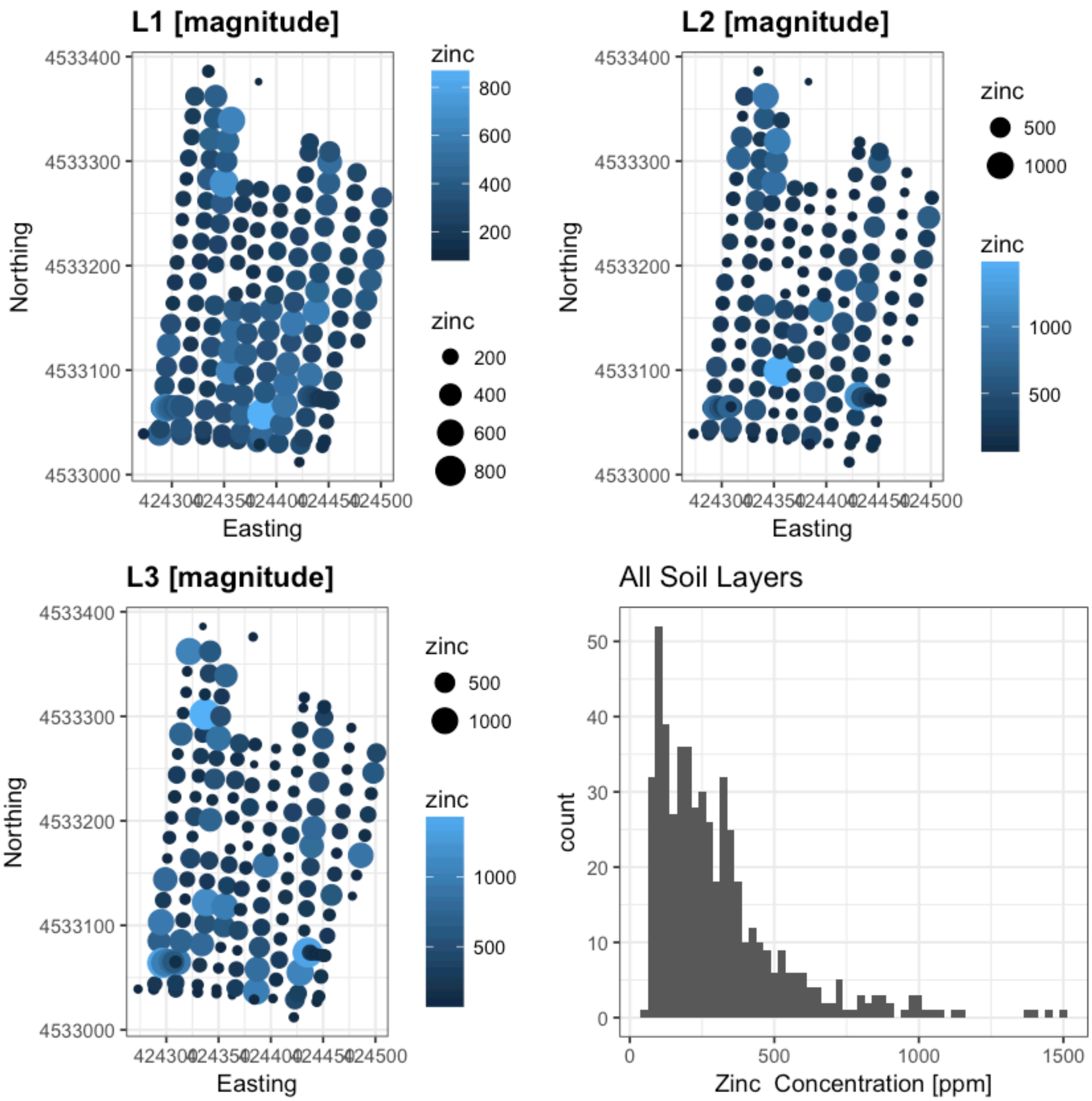
Istogramma delle concentrazioni in funzione delle profondità

```
In [42]: # S Y N T H E S I S
         tgc_Z <- summarySE( zinc,measurevar="zinc",groupvars=c("Layer","zincthre") )
         tgc_C <- summarySE( chrome,measurevar="chrome",groupvars=c("Layer","chromethre") )
         # bar :: aggregating previous response
         plt1 <- ggplot(tgc_Z, aes(x=Layer, y=zinc, fill=factor(zincthre))) +
             geom_bar(position=position_dodge(), stat="identity",
                      colour="black", # Use black outlines,
                      size=.3) +      # Thinner lines
             geom_errorbar(aes(ymin=zinc-se, ymax=zinc+se),
                           size=.3,    # Thinner lines
                           width=.2,
                           position=position_dodge(.9)) +
             xlab("Soil Layers") +
             ylab( paste("Zinc ","Concentration [ppm]")) +
             scale_fill_hue(name="Law Threshold", # Legend label, use darker colors
                            breaks=c(0, 1),
                            labels=c("Below", "Above")) +
             ggtitle("The Effect of Magnitude and Soil Layer\non Uncertainty of Chemical Concent
             #scale_y_continuous(breaks=0:20*4) +
             theme_bw()
         plt2 <- ggplot(tgc_C, aes(x=Layer, y=chrome, fill=factor(chromethre))) +
             geom_bar(position=position_dodge(), stat="identity",
                      colour="black", # Use black outlines,
                      size=.3) +      # Thinner lines
             geom_errorbar(aes(ymin=chrome-se, ymax=chrome+se),
                           size=.3,    # Thinner lines
                           width=.2,
                           position=position_dodge(.9)) +
             xlab("Soil Layers") +
             ylab( paste("Chrome ","Concentration [ppm]")) +
             scale_fill_hue(name="Law Threshold", # Legend label, use darker colors
                            breaks=c(0, 1),
                            labels=c("Below", "Above")) +
             ggtitle("The Effect of Magnitude and Soil Layer\non Uncertainty of Chemical Concent
             #scale_y_continuous(breaks=0:20*4) +
             theme_bw()
         grid.arrange(plt1,plt2,ncol=1,nrow=2)
```



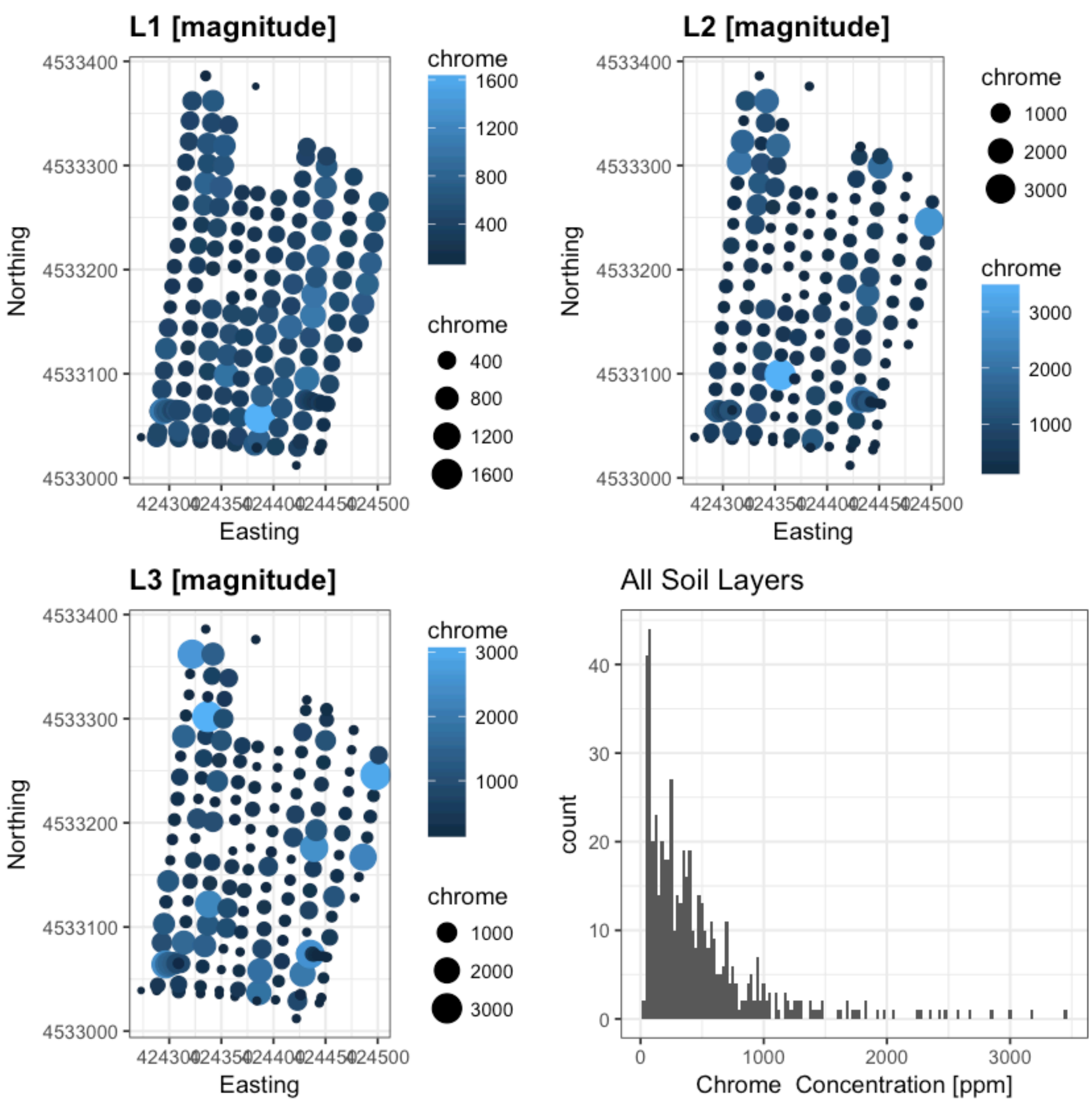
Zinc levels in soils at different depths:

```
In [43]: plt1 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==1),], colour=zinc, size=zi
         plt2 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==2),], colour=zinc, size=zi
         plt3 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==3),], colour=zinc, size=zi
         plt4 <- qplot(zinc, data=zinc, geom="histogram",binwidth=25,main="All Soil Layers") + t
         grid.arrange(plt1,plt2,plt3,plt4,ncol=2,nrow=2)
```



Chrome levels in soils at different depths:

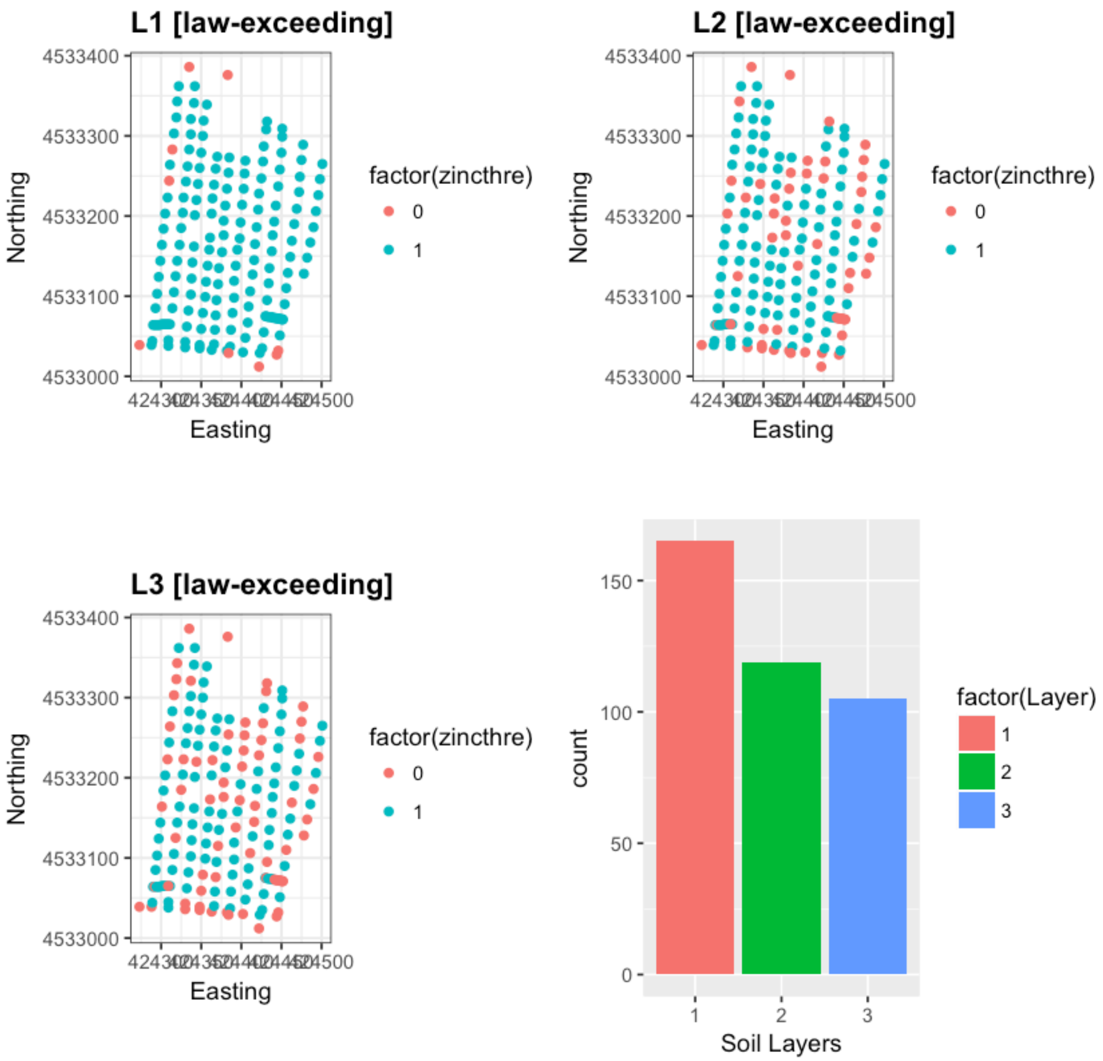
```
In [44]: plt1 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==1),], colour=chrome, s
         plt2 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==2),], colour=chrome, s
         plt3 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==3),], colour=chrome, s
         plt4 <- qplot(chrome, data=chrome, geom="histogram",binwidth=25,main="All Soil Layers")
         grid.arrange(plt1,plt2,plt3,plt4,ncol=2,nrow=2)
```



Superamento limiti di legge

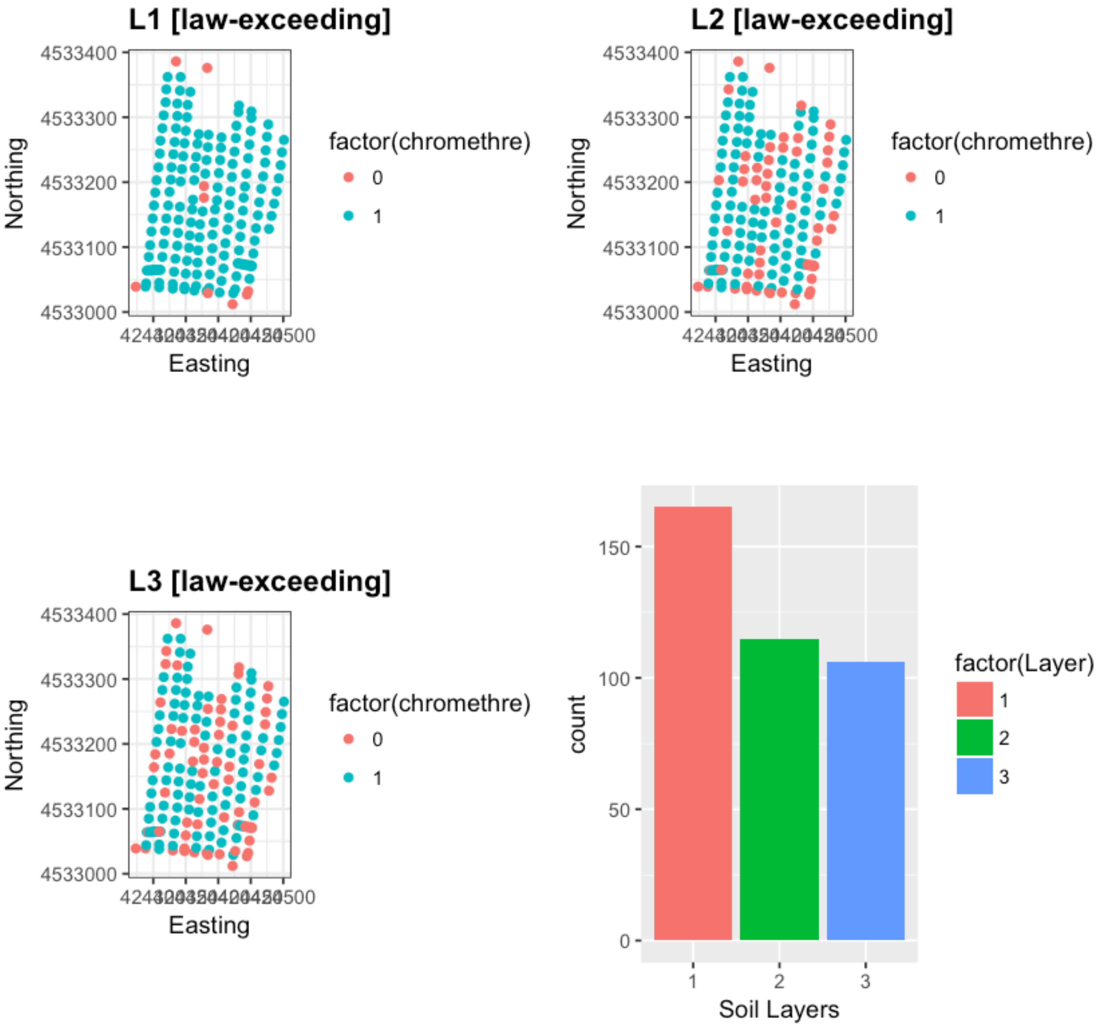
Zinc

```
In [45]: plt1 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==1),], colour=factor(zincth
         plt2 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==2),], colour=factor(zincth
         plt3 <- qplot(Easting, Northing, data=zinc[which(zinc$Layer==3),], colour=factor(zincth
         plt4 <- qplot(factor(Layer),data=zinc[which(zinc$zincthre==1),],geom="bar",fill=factor(
         grid.arrange(plt1,plt2,plt3,plt4,ncol=2,nrow=2)
```



Chrome

```
In [46]: plt1 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==1),], colour=factor(ch
         plt2 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==2),], colour=factor(ch
         plt3 <- qplot(Easting, Northing, data=chrome[which(chrome$Layer==3),], colour=factor(ch
         plt4 <- qplot(factor(Layer),data=chrome[which(chrome$chromethre==1),],geom="bar",fill=f
         grid.arrange(plt1,plt2,plt3,plt4,ncol=2,nrow=2)
```



Incertezza di misura al variare del livello di concentrazione del contaminante

Zinc

```
In [ ]: ggplot(zinc, aes(x=1:length(zinc$zinc), y=zinc, colour=factor(zincthre))) +
            geom_errorbar(aes(ymin=zinc-std, ymax=zinc+std), width=.01) +
            xlab("Samples") +
            ylab( paste("Zinc ","Concentration [ppm]") ) +
            scale_colour_hue(name="Law Threshold",          # Legend label, use darker colors
                             breaks=c(0, 1),
                             labels=c("Below", "Above"),
                             l=40) +                        # Use darker colors, lightness=40
            ggtitle("The Effect of Magnitude \n on Chemical Concentration Uncertainty") +
            expand_limits(y=0) +                         # Expand y range
            #scale_y_continuous(breaks=0:20*4) +         # Set tick every 4
            theme_bw() +
            theme(legend.justification=c(1,0)
                  #,legend.position=c(.5,.8)              # Position legend in bottom right
                  )
```

Chrome

```
In [ ]: ggplot(chrome, aes(x=1:length(chrome$chrome), y=chrome, colour=factor(chromethre))) +
            geom_errorbar(aes(ymin=chrome-std, ymax=chrome+std), width=.01) +
            xlab("Samples") +
            ylab( paste("Chrome ","Concentration [ppm]") ) +
            scale_colour_hue(name="Law Threshold",             # Legend label, use darker colors
                             breaks=c(0, 1),
                             labels=c("Below", "Above"),
                             l=40) +                     # Use darker colors, lightness=40
            ggtitle("The Effect of Magnitude \n on Chemical Concentration Uncertainty") +
            expand_limits(y=0) +                         # Expand y range
            #scale_y_continuous(breaks=0:20*4) +         # Set tick every 4
            theme_bw() +
            theme(legend.justification=c(1,0)
                  #,legend.position=c(.5,.8)              # Position legend in bottom right
                  )
```

0.1.7 Grids

Configurazione del contesto geospaziale:

```
In [57]: #___________
         #
         # G R I D S
         #___________

         # data
         coordinates(zinc)   = ~Easting+Northing
         coordinates(chrome) = ~Easting+Northing
         # grid
         coordinates(grd) = ~Easting+Northing
         gridded(grd) = TRUE
```

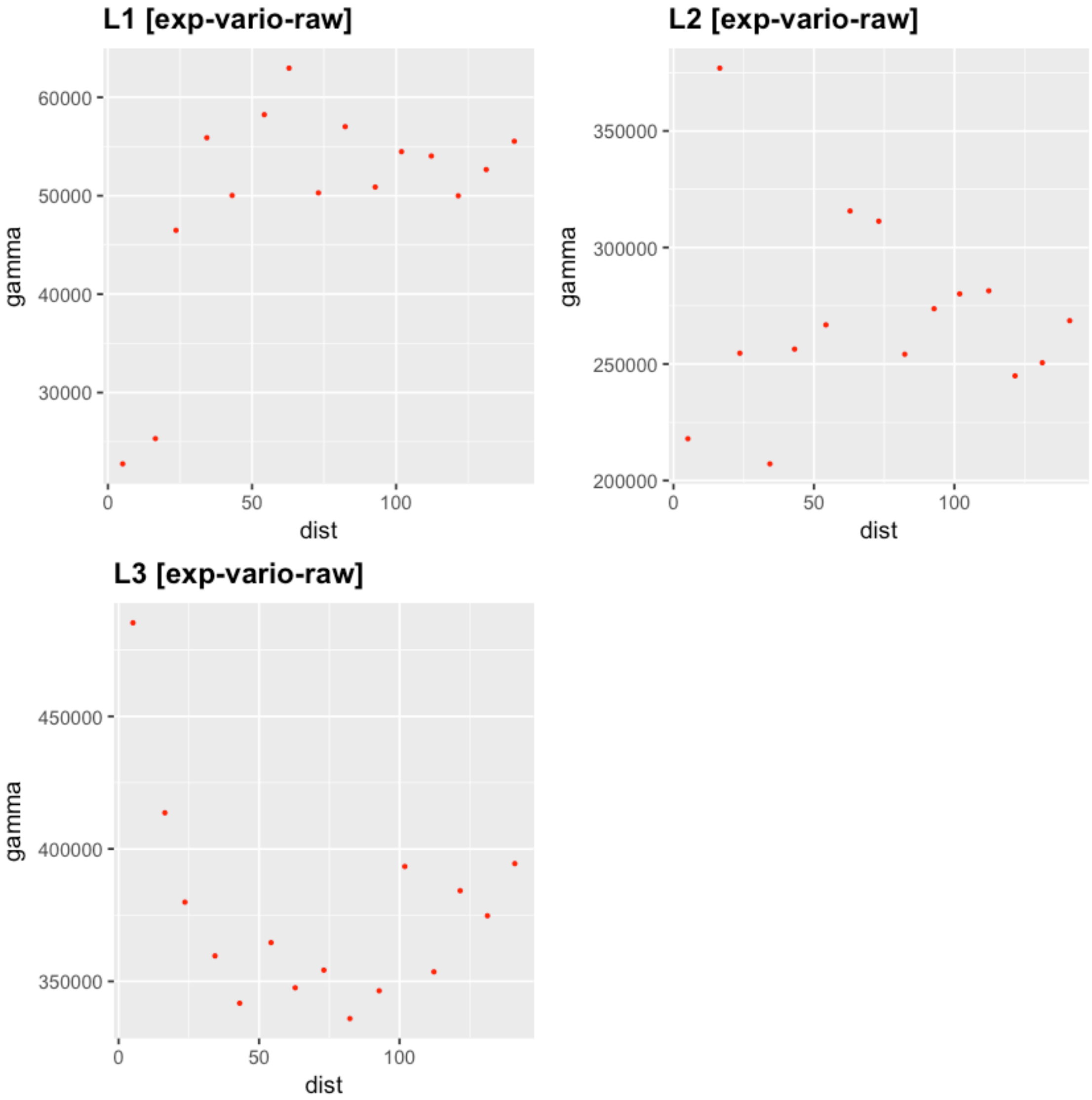
Sommario del contesto geospaziale:

```
In [59]: #summary(zinc)
         #summary(chrome)
```

0.1.8 CHROME

Variography | omnidirectional | raw

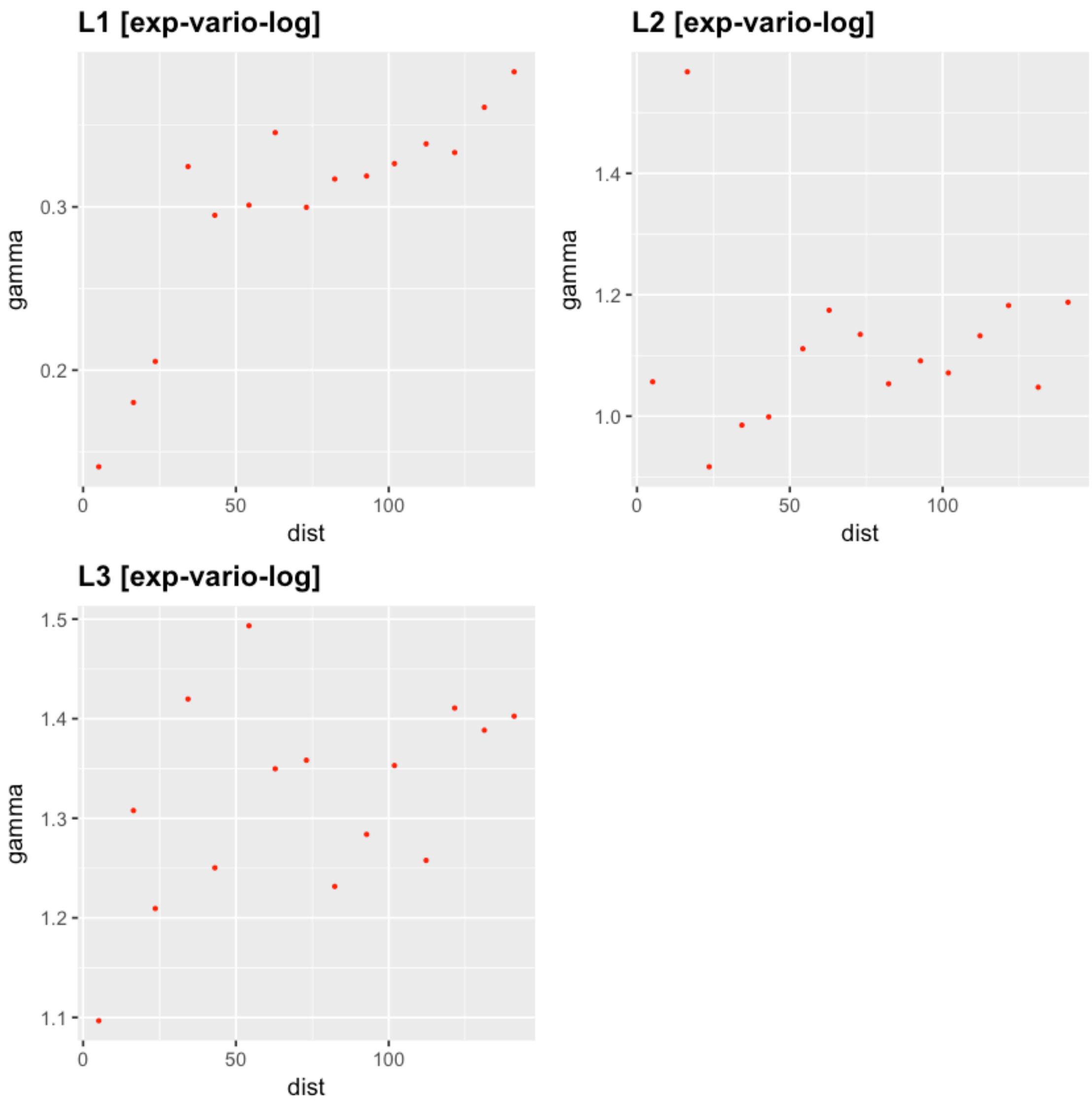
```
In [60]: vgm.exp.n.L1          = variogram( chrome~1, chrome[which(chrome$Layer==1),])
         vgm.exp.n.L2          = variogram( chrome~1, chrome[which(chrome$Layer==2),])
         vgm.exp.n.L3          = variogram( chrome~1, chrome[which(chrome$Layer==3),])
         plt1 <- ggplot(data=vgm.exp.n.L1,aes(x=dist,y=gamma)) + geom_point(color='red',size=0.5
         plt2 <- ggplot(data=vgm.exp.n.L2,aes(x=dist,y=gamma)) + geom_point(color='red',size=0.5
         plt3 <- ggplot(data=vgm.exp.n.L3,aes(x=dist,y=gamma)) + geom_point(color='red',size=0.5
         grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```



Variography | omnidirectional | log

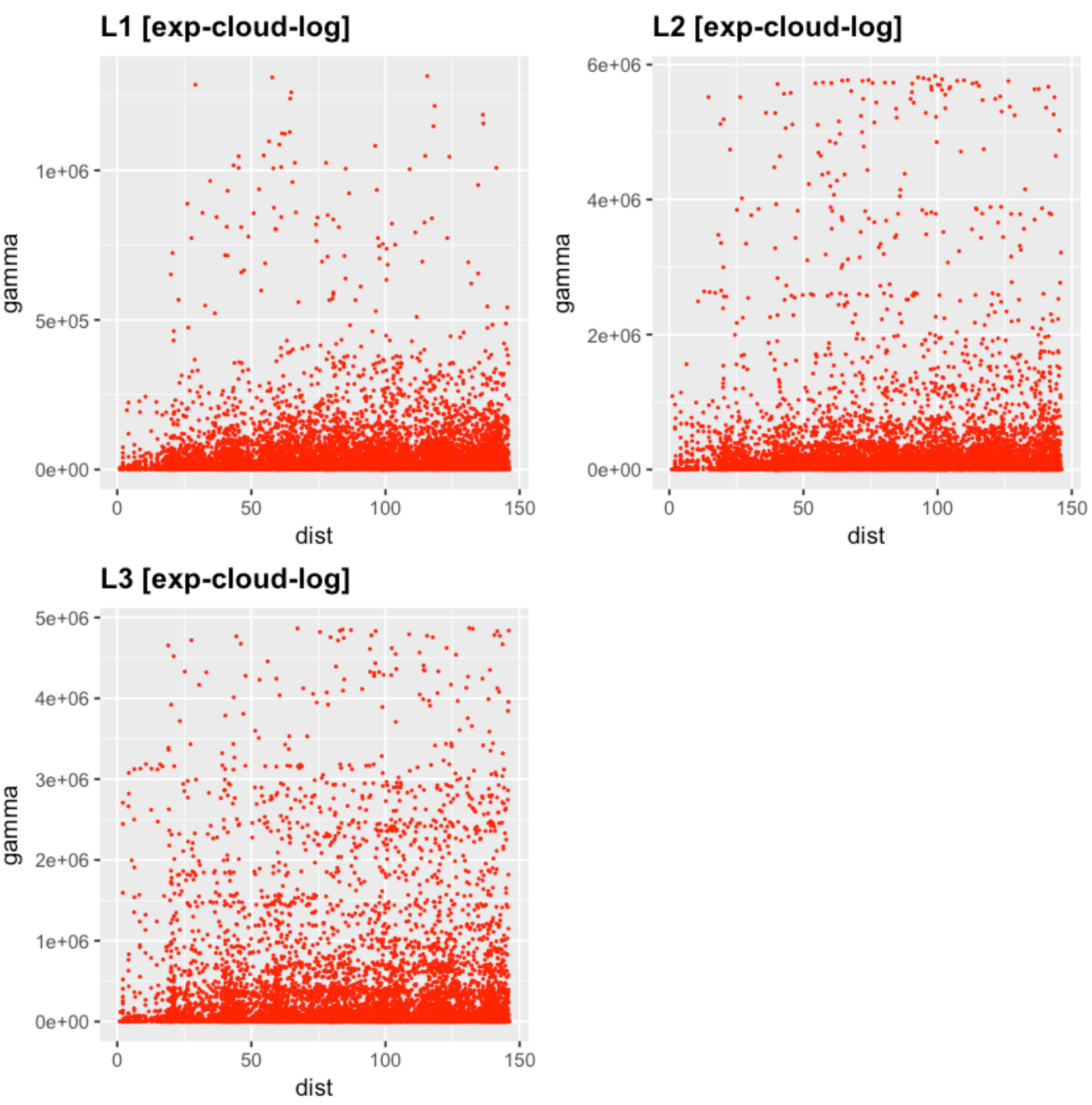
```
In [61]: vgm.exp.log.L1 = variogram( log(chrome)~1, chrome[which(chrome$Layer==1),])
         vgm.exp.log.L2 = variogram( log(chrome)~1, chrome[which(chrome$Layer==2),])
         vgm.exp.log.L3 = variogram( log(chrome)~1, chrome[which(chrome$Layer==3),])
```

```
plt1 <- ggplot(data=vgm.exp.log.L1,aes(x=dist,y=gamma)) + geom_point(color='red',size=0
plt2 <- ggplot(data=vgm.exp.log.L2,aes(x=dist,y=gamma)) + geom_point(color='red',size=0
plt3 <- ggplot(data=vgm.exp.log.L3,aes(x=dist,y=gamma)) + geom_point(color='red',size=0
grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```



Variography | omnidirectional | log & cloud

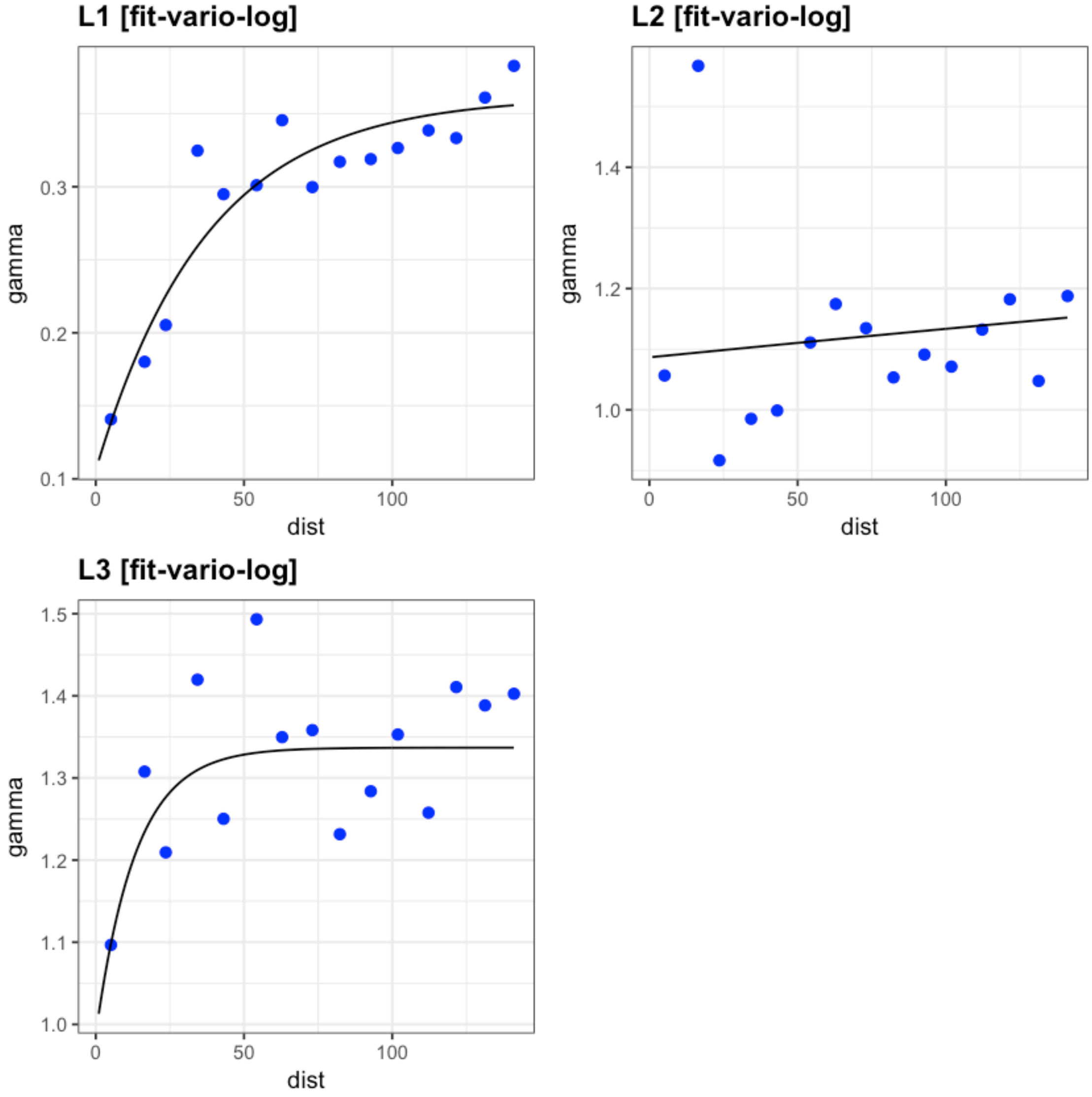
```
In [62]: vgm.exp.log.L1c         = variogram( chrome~1, chrome[which(chrome$Layer==1),], cloud=T
         vgm.exp.log.L2c         = variogram( chrome~1, chrome[which(chrome$Layer==2),], cloud=T
         vgm.exp.log.L3c         = variogram( chrome~1, chrome[which(chrome$Layer==3),], cloud=T
         plt1 <- ggplot(data=vgm.exp.log.L1c,aes(x=dist,y=gamma)) + geom_point(color='red',size=
         plt2 <- ggplot(data=vgm.exp.log.L2c,aes(x=dist,y=gamma)) + geom_point(color='red',size=
         plt3 <- ggplot(data=vgm.exp.log.L3c,aes(x=dist,y=gamma)) + geom_point(color='red',size=
         grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```



```
In [63]: #   -fitting        vgm(psill, model, range, nugget, ...)
         vgm.fit.log.L1 = fit.variogram( vgm.exp.log.L1, model=vgm(0.30,'Exp',100,0.1), fit.sill
         vgm.fit.log.L2 = fit.variogram( vgm.exp.log.L2, model=vgm(1.00,'Exp',100,0.1), fit.sill
         vgm.fit.log.L3 = fit.variogram( vgm.exp.log.L3, model=vgm(1.00,'Exp',100,0.1), fit.sill

         # L1
         maxDist <- round(max(vgm.exp.log.L1$dist))
         vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L1$psill[1] + vgm.fit.log.L1$
         names(vgm.fit.plot) <- c("dist","gamma")
         plt1 <- ggplot(data=vgm.exp.log.L1,aes(x=dist,y=gamma)) + geom_point(color='blue',size=
                                         ggtitle("L1 [fit-vario-log]") + theme(plot.title = elem
         # L2
         maxDist <- round(max(vgm.exp.log.L2$dist))
```

```
vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L2$psill[1] + vgm.fit.log.L2$
names(vgm.fit.plot) <- c("dist","gamma")
plt2 <- ggplot(data=vgm.exp.log.L2,aes(x=dist,y=gamma)) + geom_point(color='blue',size=
                                ggtitle("L2 [fit-vario-log]") + theme(plot.title = elem
# L3
maxDist <- round(max(vgm.exp.log.L3$dist))
vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L3$psill[1] + vgm.fit.log.L3$
names(vgm.fit.plot) <- c("dist","gamma")
plt3 <- ggplot(data=vgm.exp.log.L3,aes(x=dist,y=gamma)) + geom_point(color='blue',size=
                                ggtitle("L3 [fit-vario-log]") + theme(plot.title = elem
grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```

```
Warning message in fit.variogram(vgm.exp.log.L2, model = vgm(1, "Exp", 100, 0.1), :
No convergence after 200 iterations: try different initial values?
```



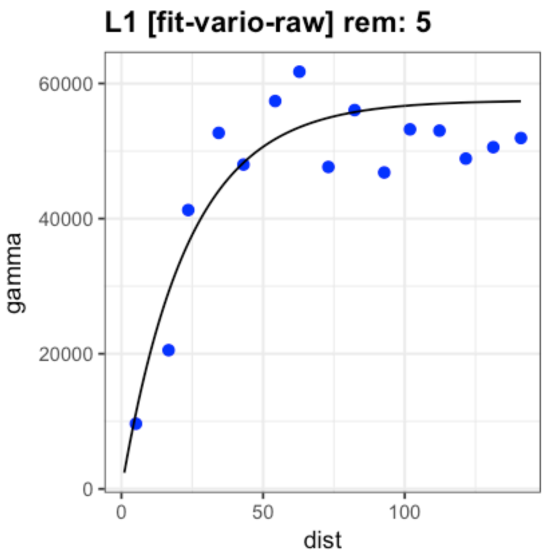
Rimozione delle coppie di punti caratterizzati da alta covarianza a corta distanza:

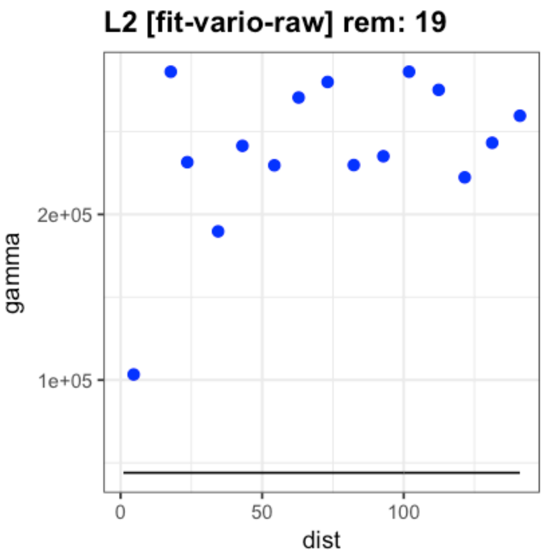
```
In [66]: # remove couples with high covariance but short distance apart:
         #   <L1>
         chrome_L1                         <- data.frame(        chrome[which(chrome$Layer==1),]
         coordinates(chrome_L1) = ~Easting+Northing
         vgm.exp.log.L1c_rd = variogram( chrome~1, chrome_L1, cloud=TRUE)
         short_dist         <- vgm.exp.log.L1c[which(vgm.exp.log.L1c_rd$dist<5.0),]
         high_gamma         <- short_dist[which(short_dist$gamma>100000),]
         high_gamma         <- data.frame(high_gamma)
         list_remove <- sort(unique( c(high_gamma$left,high_gamma$right) ))
         rem.L1 <- length(list_remove)
         if(length(list_remove)==0){
                 chrome_L1_rd   <- chrome_L1
         }else{
                 chrome_L1_rd   <- chrome_L1[-list_remove,]
         }

         #   <L2>
         chrome_L2                         <- data.frame(        chrome[which(chrome$Layer==2),]
         coordinates(chrome_L2) = ~Easting+Northing
         vgm.exp.log.L2c_rd = variogram( chrome~1, chrome_L2, cloud=TRUE)
         short_dist         <- vgm.exp.log.L2c[which(vgm.exp.log.L2c_rd$dist<5.0),]
         high_gamma         <- short_dist[which(short_dist$gamma>100000),]
         high_gamma         <- data.frame(high_gamma)
         list_remove <- sort(unique( c(high_gamma$left,high_gamma$right) ))
         rem.L2 <- length(list_remove)
         if(length(list_remove)==0){
                 chrome_L2_rd   <- chrome_L2
         }else{
                 chrome_L2_rd   <- chrome_L2[-list_remove,]
         }

         #   <L3>
         chrome_L3                         <- data.frame(        chrome[which(chrome$Layer==3),]
         coordinates(chrome_L3) = ~Easting+Northing
         vgm.exp.log.L3c_rd = variogram( chrome~1, chrome_L3, cloud=TRUE)
         short_dist         <- vgm.exp.log.L3c[which(vgm.exp.log.L3c_rd$dist<5.0),]
         high_gamma         <- short_dist[which(short_dist$gamma>100000),]
         high_gamma         <- data.frame(high_gamma)
         list_remove <- sort(unique( c(high_gamma$left,high_gamma$right) ))
         chrome_L3_rd   <- chrome_L3[-list_remove,]
         rem.L3 <- length(list_remove)
         if(length(list_remove)==0){
                 chrome_L3_rd   <- chrome_L3
         }else{
                 chrome_L3_rd   <- chrome_L3[-list_remove,]
         }
```

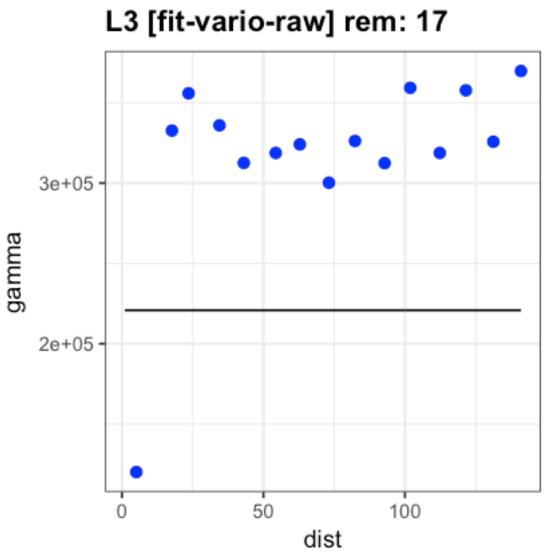
REMOVED [fit-vario-raw]:

```
In [67]: #coordinates(xrf_L1_rd) = ~Easting+Northing
         vgm.exp.raw.L1_rd = variogram( chrome~1, chrome_L1_rd)# vgm( psill, model, range, nugge
         psill         = max(vgm.exp.raw.L1_rd$gamma) - min(vgm.exp.raw.L1_rd$gamma)
         range         = max(vgm.exp.raw.L1_rd$dist)/6
         nugget         = vgm.exp.raw.L1_rd$gamma[1]/2
         vgm.fit.raw.L1_rd = fit.variogram( vgm.exp.raw.L1_rd, model=vgm(psill,'Exp',range,nugge
         #coordinates(xrf_L2_rd) = ~Easting+Northing
         vgm.exp.raw.L2_rd = variogram( chrome~1, chrome_L2_rd)
         psill         = max(vgm.exp.raw.L2_rd$gamma) - min(vgm.exp.raw.L2_rd$gamma)
         range         = max(vgm.exp.raw.L2_rd$dist)/6
         nugget         = vgm.exp.raw.L2_rd$gamma[1]/2
         vgm.fit.raw.L2_rd = fit.variogram( vgm.exp.raw.L2_rd, model=vgm(psill,'Exp',range,nugge
         #coordinates(xrf_L3_rd) = ~Easting+Northing
         vgm.exp.raw.L3_rd = variogram( chrome~1, chrome_L3_rd)
         psill         = max(vgm.exp.raw.L3_rd$gamma) - min(vgm.exp.raw.L3_rd$gamma)
         range         = max(vgm.exp.raw.L3_rd$dist)/6
         nugget         = vgm.exp.raw.L3_rd$gamma[1]
         vgm.fit.raw.L3_rd = fit.variogram( vgm.exp.raw.L3_rd, model=vgm(psill,'Exp',range,nugge
         #plot(vgm.exp_log,vgm.fit_log, main = paste("[OMNI-DIR] Fitted Variogram",ChElstr,"[rau
         # L1
         maxDist <- round(max(vgm.exp.raw.L1_rd$dist))
         vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.raw.L1_rd$psill[1] + vgm.fit.raw.
         names(vgm.fit.plot) <- c("dist","gamma")
         plt1 <- ggplot(data=vgm.exp.raw.L1_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                         ggtitle(paste("L1 [fit-vario-raw] rem:",rem.L1)) + them
         # L2
         maxDist <- round(max(vgm.exp.raw.L2_rd$dist))
         vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.raw.L2_rd$psill[1] + vgm.fit.raw.
         names(vgm.fit.plot) <- c("dist","gamma")
         plt2 <- ggplot(data=vgm.exp.raw.L2_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                         ggtitle(paste("L2 [fit-vario-raw] rem:",rem.L2)) + them
         # L3
         maxDist <- round(max(vgm.exp.raw.L3_rd$dist))
         vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.raw.L3_rd$psill[1] + vgm.fit.raw.
         names(vgm.fit.plot) <- c("dist","gamma")
         plt3 <- ggplot(data=vgm.exp.raw.L3_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                         ggtitle(paste("L3 [fit-vario-raw] rem:",rem.L3)) + them
         grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```

```
Warning message in fit.variogram(vgm.exp.raw.L1_rd, model = vgm(psill, "Exp", range, :
partial sill or nugget fixed at zero valueWarning message in fit.variogram(vgm.exp.raw.L2_rd, mo
singular model in variogram fitWarning message in fit.variogram(vgm.exp.raw.L3_rd, model = vgm(p
No convergence after 200 iterations: try different initial values?Warning message in fit.variogr
partial sill or nugget fixed at zero valueWarning message in fit.variogram(object, model, fit.si
singular model in variogram fit
```


L1 [fit-vario-raw] rem: 5

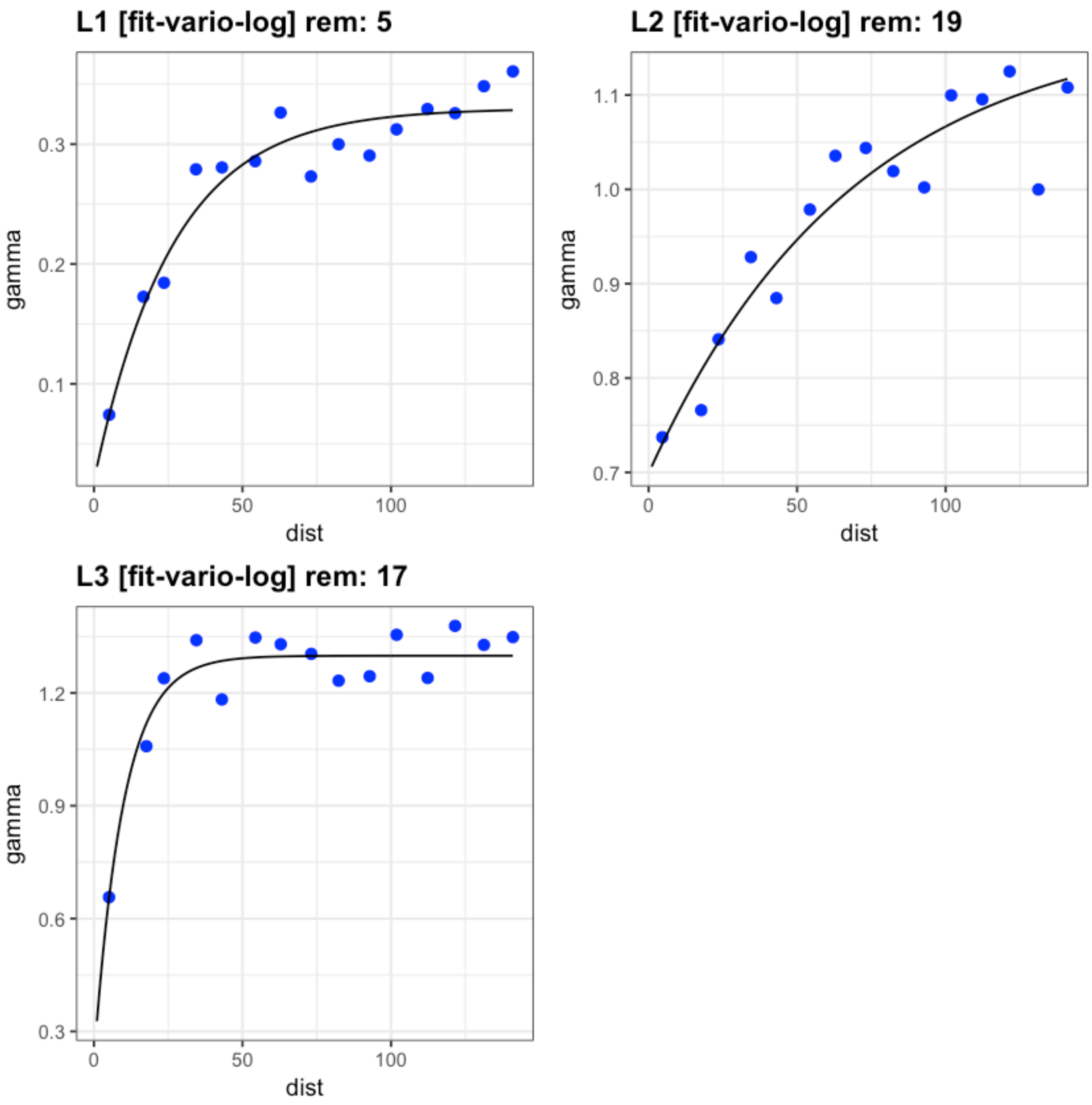

L2 [fit-vario-raw] rem: 19


L3 [fit-vario-raw] rem: 17

REMOVED [fit-vario-log]:

```
In [68]: #coordinates(xrf_L1_rd) = ~Easting+Northing
         vgm.exp.log.L1_rd = variogram( log(chrome)~1, chrome_L1_rd)
         vgm.fit.log.L1_rd = fit.variogram( vgm.exp.log.L1_rd, model=vgm(0.90,'Exp',100,0.1), fi
         #coordinates(xrf_L2_rd) = ~Easting+Northing
         vgm.exp.log.L2_rd = variogram( log(chrome)~1, chrome_L2_rd)
         vgm.fit.log.L2_rd = fit.variogram( vgm.exp.log.L2_rd, model=vgm(0.90,'Exp',100,0.1), fi
         #coordinates(xrf_L3_rd) = ~Easting+Northing
         vgm.exp.log.L3_rd = variogram( log(chrome)~1, chrome_L3_rd)
         vgm.fit.log.L3_rd = fit.variogram( vgm.exp.log.L3_rd, model=vgm(0.90,'Exp',100,0.1), fi
         #plot(vgm.exp_log,vgm.fit_log, main = paste("[OMNI-DIR] Fitted Variogram",ChElstr,"[log
         # L1
```

```
maxDist <- round(max(vgm.exp.log.L1_rd$dist))
vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L1_rd$psill[1] + vgm.fit.log.
names(vgm.fit.plot) <- c("dist","gamma")
plt1 <- ggplot(data=vgm.exp.log.L1_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                ggtitle(paste("L1 [fit-vario-log] rem:",rem.L1)) + them
# L2
maxDist <- round(max(vgm.exp.log.L2_rd$dist))
vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L2_rd$psill[1] + vgm.fit.log.
names(vgm.fit.plot) <- c("dist","gamma")
plt2 <- ggplot(data=vgm.exp.log.L2_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                ggtitle(paste("L2 [fit-vario-log] rem:",rem.L2)) + them
# L3
maxDist <- round(max(vgm.exp.log.L3_rd$dist))
vgm.fit.plot <- data.frame(1:maxDist,g_exp <- vgm.fit.log.L3_rd$psill[1] + vgm.fit.log.
names(vgm.fit.plot) <- c("dist","gamma")
plt3 <- ggplot(data=vgm.exp.log.L3_rd,aes(x=dist,y=gamma)) + geom_point(color='blue',si
                                ggtitle(paste("L3 [fit-vario-log] rem:",rem.L3)) + them
grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```



0.1.9 Predictions

```
In [84]: #printf( "Reference system: %s\n",proj4string(grd) )
         #printf( "Pixel size [%fm, %fm]\n", res(grd)[1],res(grd)[2] ) # x-y pixel size
         #plot(grd)#,add=T) # add monitoring points
```

Assign projection to point data

```
In [ ]: #proj4string(zinc)   <- CRS(proj4string(grd))
        #proj4string(chrome) <- CRS(proj4string(grd))
```

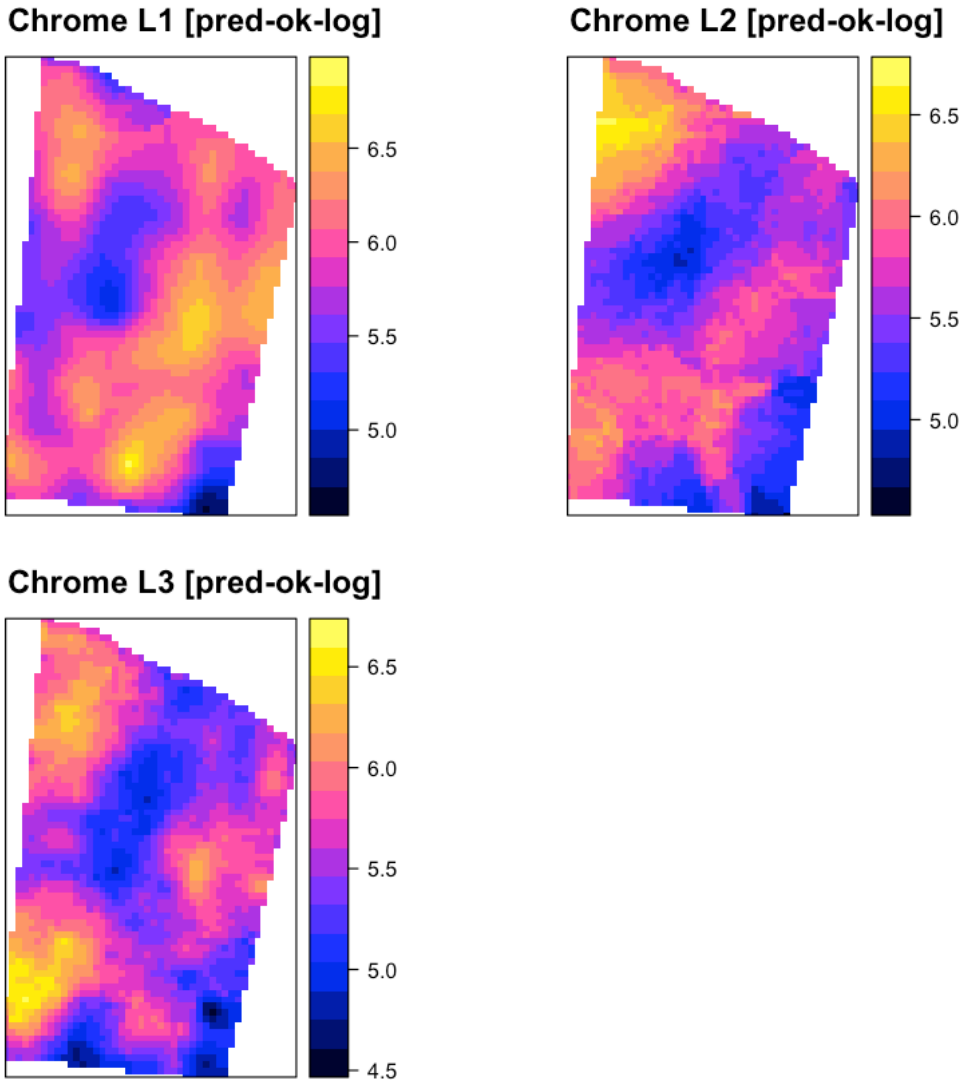
Ordinary Kriging   R A W

```
In [ ]: # ...
```

LOG

```
In [85]: #   -local
         krg.ok.log.L1 = krige(log(chrome)~1, chrome[which(chrome$Layer==1),], grd, model = vgm.
         krg.ok.log.L2 = krige(log(chrome)~1, chrome[which(chrome$Layer==2),], grd, model = vgm.
         krg.ok.log.L3 = krige(log(chrome)~1, chrome[which(chrome$Layer==3),], grd, model = vgm.
         plt1 = spplot(krg.ok.log.L1, "var1.pred", main="Chrome L1 [pred-ok-log]")
         plt2 = spplot(krg.ok.log.L2, "var1.pred", main="Chrome L2 [pred-ok-log]")
         plt3 = spplot(krg.ok.log.L3, "var1.pred", main="Chrome L3 [pred-ok-log]")
         grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```

```
[using ordinary kriging]
[using ordinary kriging]
[using ordinary kriging]
```



RAW | After Removal

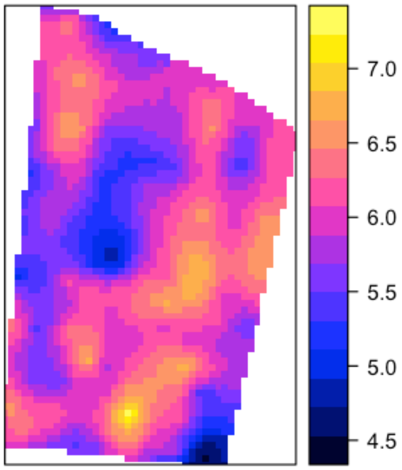
```
In [ ]: # ...
```

LOG | After Removal

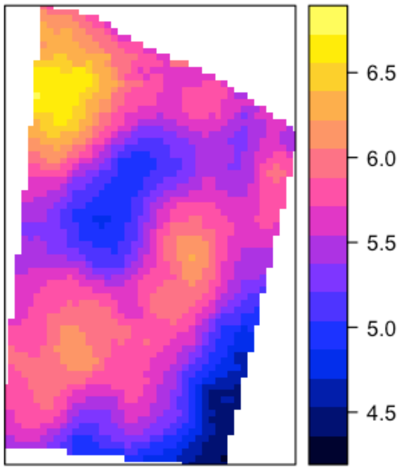
```
In [86]: krg.ok.log.L1_rd = krige(log(chrome)~1, chrome_L1_rd, grd, model = vgm.fit.log.L1_rd, n
         krg.ok.log.L2_rd = krige(log(chrome)~1, chrome_L2_rd, grd, model = vgm.fit.log.L2_rd, n
         krg.ok.log.L3_rd = krige(log(chrome)~1, chrome_L3_rd, grd, model = vgm.fit.log.L3_rd, n
         plt1 = spplot(krg.ok.log.L1_rd, "var1.pred", main="Chrome L1 [pred-ok-log] adj")
         plt2 = spplot(krg.ok.log.L2_rd, "var1.pred", main="Chrome L2 [pred-ok-log] adj")
         plt3 = spplot(krg.ok.log.L3_rd, "var1.pred", main="Chrome L3 [pred-ok-log] adj")
         grid.arrange(plt1,plt2,plt3,ncol=2,nrow=2)
```

```
[using ordinary kriging]
[using ordinary kriging]
[using ordinary kriging]
```

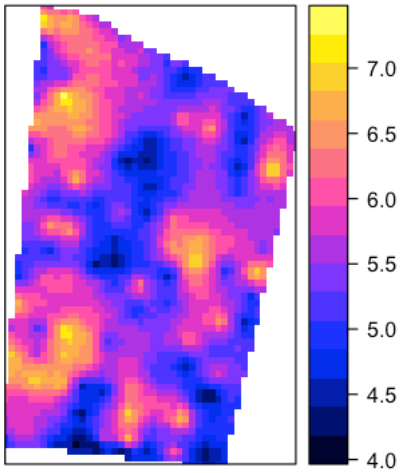
Chrome L1 [pred-ok-log] adj



Chrome L2 [pred-ok-log] adj



Chrome L3 [pred-ok-log] adj



Sintesi:
rimozione delle coppie di punti "outliers",
fitting del variogramma sperimentale con un modello di variogramma ammissibile,

```
<li> sia sulla variabile (=Cromo) tal quale, </li>
<li> sia sulla variabile (=Cromo) in scale logaritmica, </li>
```

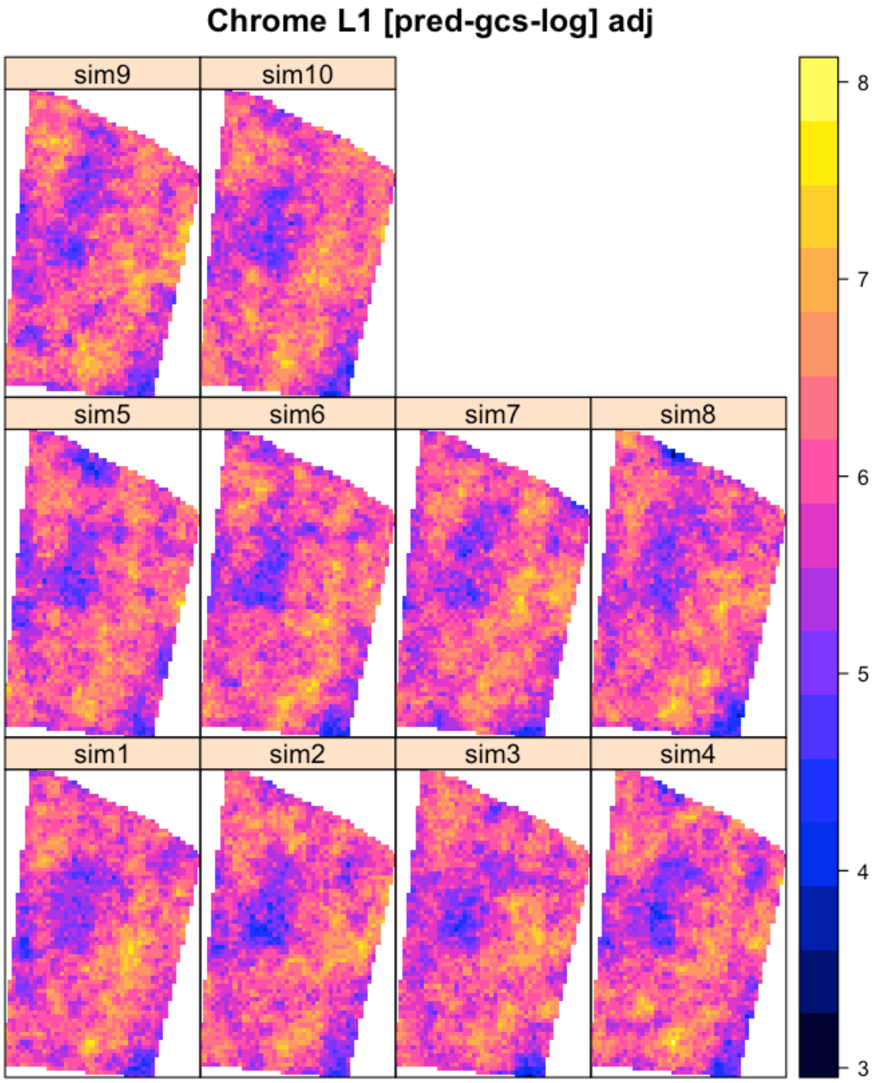
interpolazione mediante kriging ordinario in entrambi i casi,
le mappe ottenute "dopo rimozione delle coppie di punti",

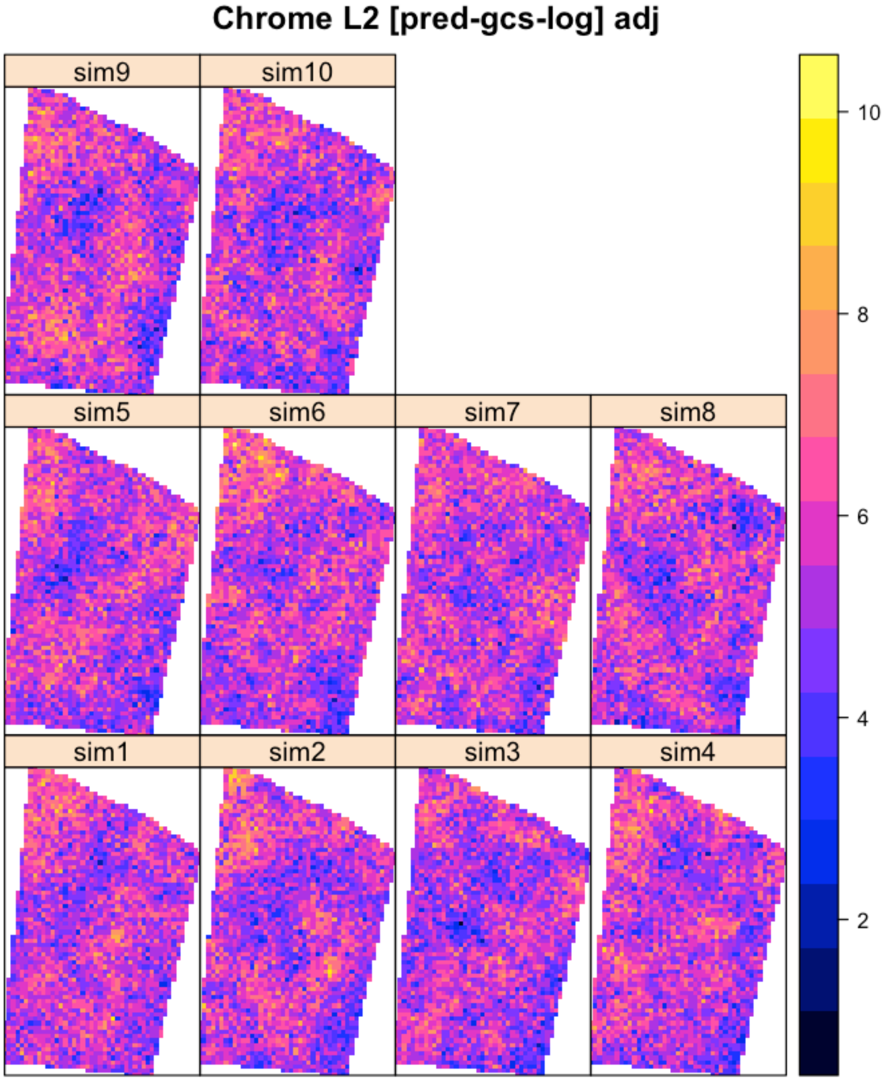
```
<li> presentano un maggiore dettaglio spaziale, </li>
<li> sono caratterizzare da minore rumore. </li>
```

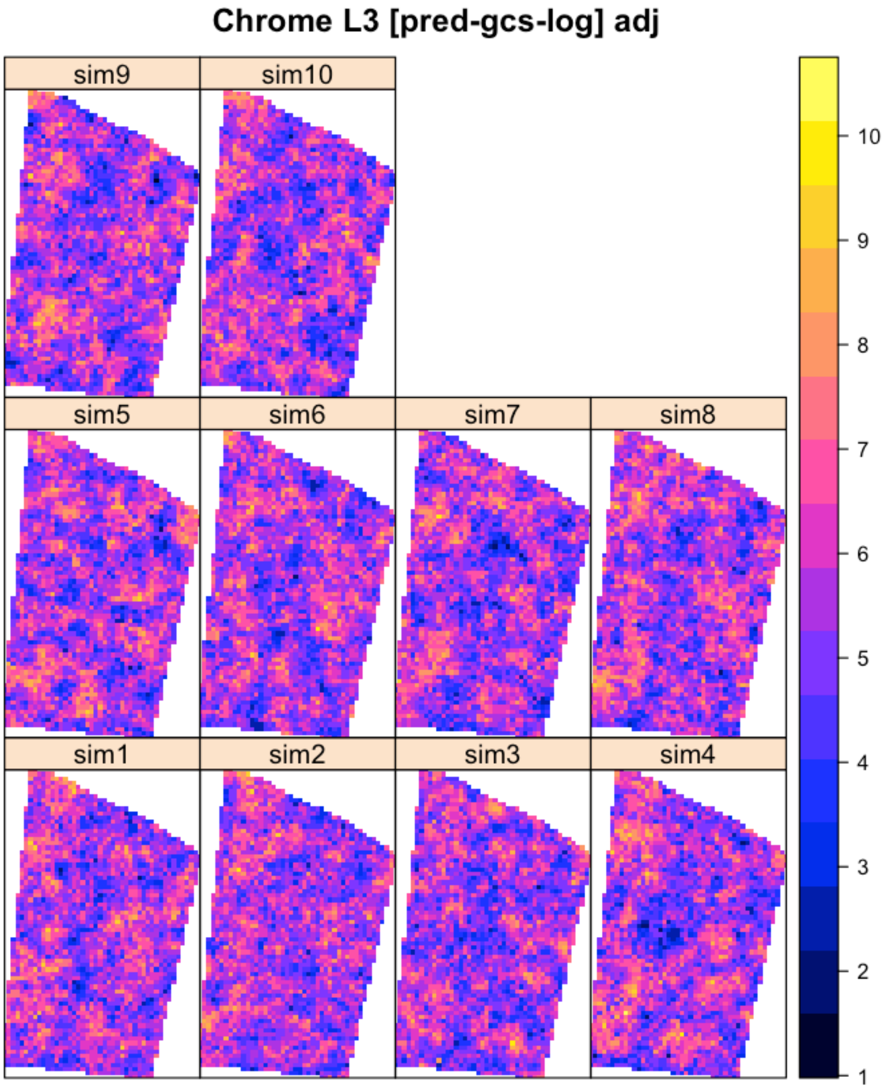
0.1.10 Conditional Gaussian Simulations

```
In [88]: gcs_nsim = 10
         krg.gcs.log.L1_rd = krige(log(chrome)~1, chrome_L1_rd, grd, model = vgm.fit.log.L1_rd,
         krg.gcs.log.L2_rd = krige(log(chrome)~1, chrome_L2_rd, grd, model = vgm.fit.log.L2_rd,
         krg.gcs.log.L3_rd = krige(log(chrome)~1, chrome_L3_rd, grd, model = vgm.fit.log.L3_rd,
         spplot(krg.gcs.log.L1_rd, main = "Chrome L1 [pred-gcs-log] adj" )
         spplot(krg.gcs.log.L2_rd, main = "Chrome L2 [pred-gcs-log] adj" )
         spplot(krg.gcs.log.L3_rd, main = "Chrome L3 [pred-gcs-log] adj" )
```

```
drawing 10 GLS realisations of beta...
[using conditional Gaussian simulation]
drawing 10 GLS realisations of beta...
[using conditional Gaussian simulation]
drawing 10 GLS realisations of beta...
[using conditional Gaussian simulation]
```




Chrome L2 [pred-gcs-log] adj
sim9
sim10
sim5
sim6
sim7
sim8
sim1
sim2
sim3
sim4
10
8
6
4
2



SAVE | Analysis of Uncertainty

```
In [90]: # gcs | layer #1
         gcs_nsim = 200
         krg.gcs.raw.L1_rd = krige(chrome~1, chrome_L1_rd, grd, model = vgm.fit.raw.L1_rd, nmax=
         write.table( krg.gcs.raw.L1_rd, file="L1_gcs_raw.txt",sep=",",na="NaN",col.names=TRUE,r
```

```
drawing 200 GLS realisations of beta...
[using conditional Gaussian simulation]
```