

Esercitazione sui dati geostatistici

I dati

I dati si riferiscono al livello medio di polveri sottili (PM10, $\mu\text{g}/\text{m}^3$) rilevato dalla rete di centraline della Regione Lombardia nel 2007.

Il file `centraline.txt` contiene:

x, y: latitudine e longitudine delle centraline (sistema geodetico di riferimento ED50)

pm10: concentrazione media di PM10

Il file `lombardia.shp` è lo shapefile dei confini regionali di Regione Lombardia (sistema geodetico di riferimento ED50, fonte: ISTAT).

1. La preparazione dei dati

1.1 Caricare le librerie necessarie per l'analisi.

```
library(spdep)#carica anche la libreria sp
library(gstat)
library(rgdal)
library(RColorBrewer)#utile per la gestione dei colori?
library(classInt)
```

1.2 Definire i colori da utilizzare nei diagrammi a bolle.

Nei diagrammi a bolle le singole osservazioni possono essere (parzialmente) sovrapposte. Per migliorare la lettura del diagramma è meglio utilizzare colori trasparenti, che è necessario definire.

```
mycolb <- rgb(0,0,255, alpha = 125,max=255)#blu
mycolr <- rgb(255,0,0, alpha = 125,max=255)#rosso
mycolg <- rgb(0,255,0, alpha = 125,max=255)#verde
mycolc <- rgb(255,165,0, alpha = 125,max=255)#magenta
```

1.3 Caricare i dati.

```
data<-read.table("centraline.txt",header=T)
data[1:10,]#prime dieci righe del dataset
```

```
##           x           y  pm10
## 1  481135.1 5051972 45.58
## 2  484311.2 5056209 45.92
## 3  487170.0 5047917 43.81
## 4  489134.7 5019419 35.11
## 5  490716.9 5034527 47.10
## 6  492543.6 4994570 33.49
## 7  497470.3 5070236 37.33
## 8  500747.4 4983100 38.63
## 9  503205.2 5052718 41.83
## 10 505285.4 5073698 40.85
```

1.4 Definire i dati come SpatialPointsDataFrame.

```
coordinates(data)<- ~ x+y
```

La funzione `coordinates` della libreria `sp` si assegna un sistema di coordinate ad un `data.frame` preesistente, creando un oggetto di classe `SpatialPointsDataFrame`. Le coordinate devono essere già incluse nel `data.frame`.

1.5 Trasformare le concentrazioni di PM10 utilizzando il logaritmo.

Generalmente le concentrazioni seguono una distribuzione log-normale, per cui è preferibile trasformarle utilizzando il logaritmo.

```
data$logpm10<- log(data$pm10)
```

1.6 Importare lo shapefile dei confini regionali e dei capoluoghi di provincia della Lombardia.

```
lombardia <- readOGR("lombardia.shp")
```

```
## Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI, dumpSRS =  
## dumpSRS, : Discarded datum European_Datum_1950 in Proj4 definition: +proj=utm  
## +zone=32 +ellps=intl +units=m +no_defs
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "C:\Users\corra\OneDrive - unige.it\didat\master_padova\kriging\tutorial\lombardia\lombardia  
## with 1 features  
## It has 4 fields
```

```
capoluoghi <- readOGR("capoluoghi_lombardia.shp")
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "C:\Users\corra\OneDrive - unige.it\didat\master_padova\kriging\tutorial\lombardia\capoluoghi  
## with 12 features  
## It has 1 fields
```

```
#definizione dei layer da aggiungere alla mappa della regione
```

```
l1<-list(capoluoghi,pch=16,col=2)
```

```
l2<-list("sp.text", coordinates(capoluoghi), capoluoghi$com)
```

```
#mappa della regione
```

```
spplot(lombardia,1,sp.layout=list(l1,l2),colorkey=F,col.regions="white")
```



La libreria **rgdal** consente la lettura e la scrittura di shapefile mediante le funzioni **readOGR** e **writeOGR**. La funzione **splot** consente di rappresentare graficamente dati di classe **sp** (punti, linee, poligoni, ecc.). In questo caso l'oggetto **lombardia** è di classe **SpatialPolygonsDataFrame**. L'argomento **sp.layout** consente di aggiungere al diagramma ulteriori layer, in questo caso i centroidi dei capoluoghi di provincia (11) ed il loro nome (12).

2. Analisi descrittiva

Prima di procedere con i metodi di analisi spaziale è bene utilizzare metodi descrittivi per capire le caratteristiche della variabile PM10 e la sua distribuzione nello spazio.

2.1 Calcolare le statistiche di sintesi della variabile PM10.

```
summary(data)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##      min      max
## x 481135.1 680872.4
## y 4983099.8 5147505.8
## Is projected: NA
## proj4string : [NA]
## Number of points: 58
## Data attributes:
##      pm10      logpm10
## Min.    :20.02  Min.    :2.997
## 1st Qu.:38.08  1st Qu.:3.640
```

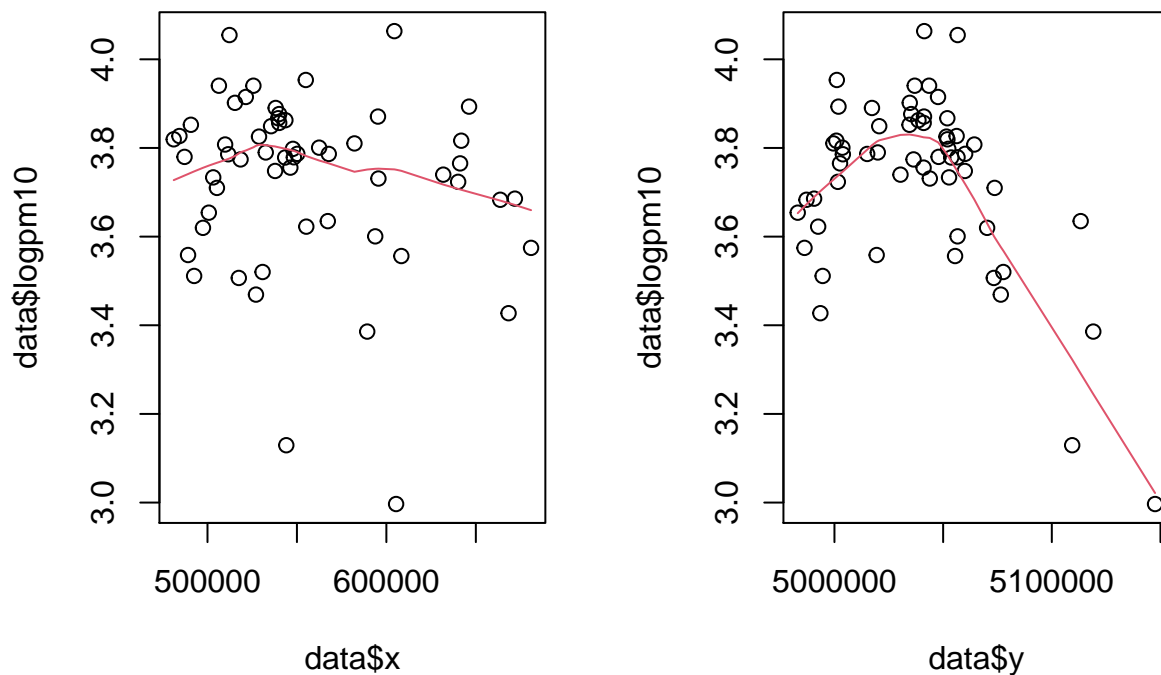
```
## Median :43.79   Median :3.779
## Mean   :42.43   Mean   :3.731
## 3rd Qu.:47.06   3rd Qu.:3.851
## Max.   :58.18   Max.   :4.064
```

Il livello medio di PM10 misurato dalle 58 centraline lombarde varia tra 20 e 58 $\mu\text{g}/\text{m}^3$, con una media pari a circa 42.

2.2 Valutare la presenza di trend spaziale.

Il diagramma di dispersione della variabile osservata rispetto alle due coordinate consente di avere una veloce indicazione sull'andamento del fenomeno studiato nello spazio.

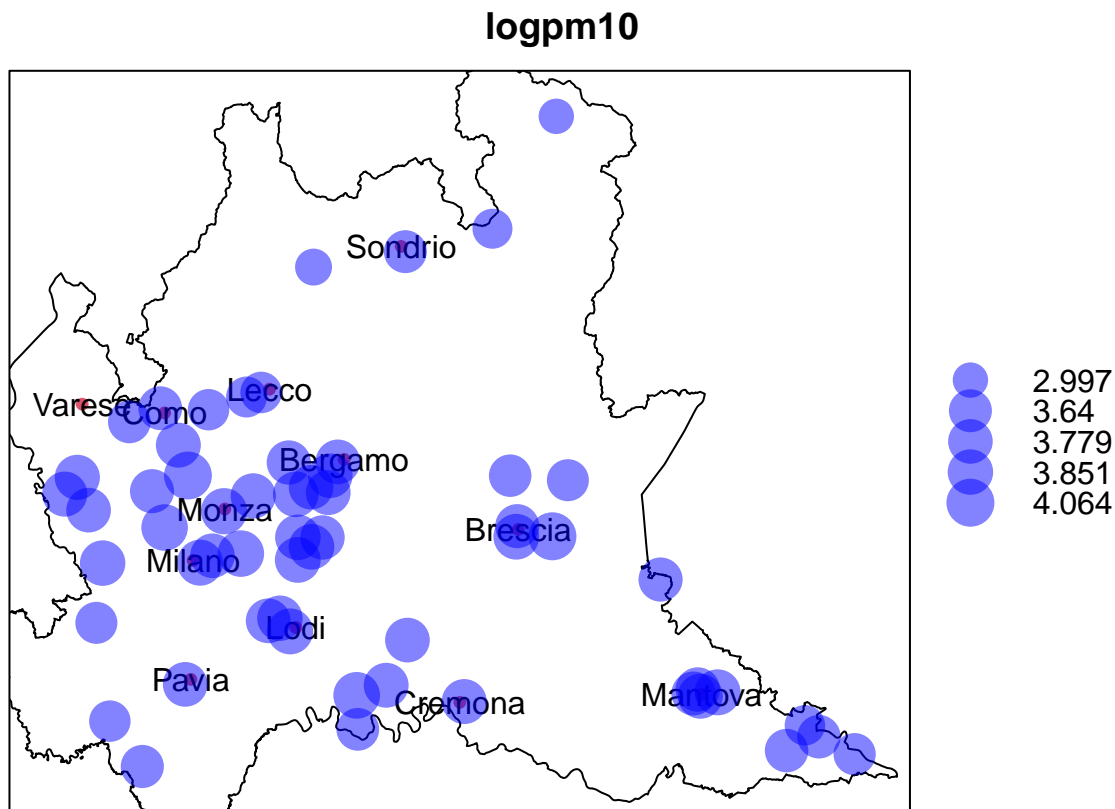
```
par(mfrow=c(1,2))
plot(data$x,data$logpm10)
lines(lowess(data$x,data$logpm10),col=2)
plot(data$y,data$logpm10)
lines(lowess(data$y,data$logpm10),col=2)
```



La linea aggiunta al diagramma di dispersione è una stima non parametrica della funzione che lega i dati alle coordinate spaziali. In caso di assenza di relazione la linea dovrebbe essere orizzontale. Mentre rispetto alla longitudine la relazione non è particolarmente accentuata (diagramma a sinistra), emerge chiaramente un forte gradiente Nord-Sud (diagramma a destra). In particolare, le centraline a Nord della regione presentano decisamente valori più bassi.

2.3 Costruire il diagramma a bolle del PM10.

```
l3<-list(lombardia)
bubble(data, "logpm10", col = c(mycolr, mycolb),sp.layout=list(l3,l1,l2),do.sqrt=F)
```



In primo luogo si nota che la distribuzione delle centraline non è regolare, esse sono infatti addensate nelle zone più urbanizzate, in particolare nella parte occidentale della regione. Anche se non è evidentissimo, si nota inoltre che le centraline situate nella zona alpina presentano valori più piccoli rispetto a quelli in prossimità dei principali centri urbani.

3. Le superfici di risposta

Il trend spaziale evidenziato dai diagrammi costruiti in precedenza può essere modellizzato mediante una superficie di risposta lineare o quadratica. Le superfici di risposta possono essere costuite mediante regressione lineare semplice.

3.1 Costruire una superficie di risposta lineare

```
x<-data$x
y<-data$y
z<-data$logpm10
linear.fit<-lm(z~x+y)
summary(linear.fit)
```

```
##
## Call:
## lm(formula = z ~ x + y)
##
## Residuals:
```

| | Min | 1Q | Median | 3Q | Max |
|----|----------|----------|---------|---------|---------|
| ## | -0.42084 | -0.06450 | 0.03262 | 0.10569 | 0.39789 |

```
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.857e+01  3.451e+00   5.381 1.57e-06 ***
## x           -1.207e-06  4.368e-07  -2.763  0.00778 **
## y           -2.811e-06  6.689e-07  -4.203  9.76e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1686 on 55 degrees of freedom
## Multiple R-squared:  0.2683, Adjusted R-squared:  0.2417
## F-statistic: 10.09 on 2 and 55 DF,  p-value: 0.0001855
```

Occorre prestare attenzione all'interpretazione dei risultati, perché in questo caso i dati non sono indipendenti e quindi l'analisi di regressione può essere interpretata solo in un'ottica descrittiva. Per valutare l'esistenza del trend è opportuno osservare anche il variogramma empirico e valutare se, dopo aver rimosso il trend spaziale, i residui diventano stazionari.

3.2 Costruire una superficie di risposta quadratica

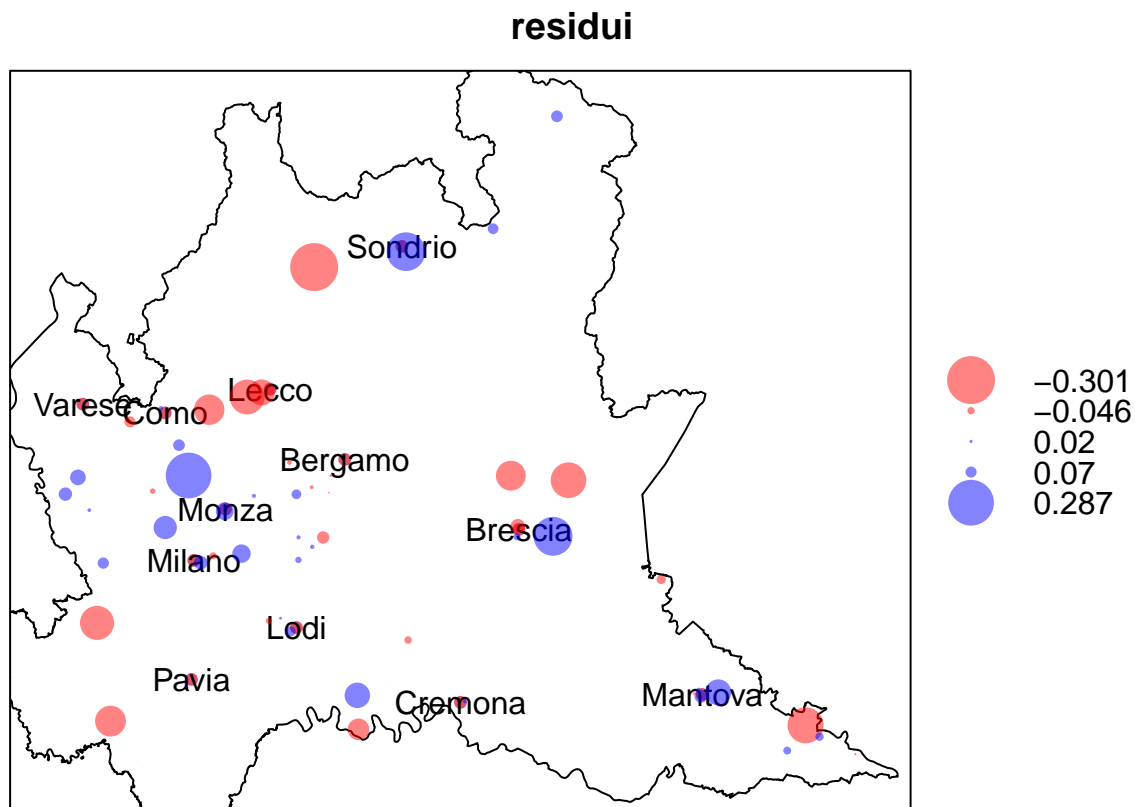
```
quadr.fit<-lm(z~x+y+I(x^2)+I(y^2)+x*y)
summary(quadr.fit)
```

```
##
## Call:
## lm(formula = z ~ x + y + I(x^2) + I(y^2) + x * y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30136 -0.04619  0.02048  0.07006  0.28723
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.626e+03  3.082e+02  -5.276 2.60e-06 ***
## x            2.072e-05  8.198e-05   0.253   0.801
## y            6.459e-04  1.264e-04   5.110 4.68e-06 ***
## I(x^2)       -1.117e-11  8.547e-12  -1.307   0.197
## I(y^2)       -6.412e-11  1.294e-11  -4.954 8.10e-06 ***
## x:y          -1.606e-12  1.498e-11  -0.107   0.915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1275 on 52 degrees of freedom
## Multiple R-squared:  0.6043, Adjusted R-squared:  0.5662
## F-statistic: 15.88 on 5 and 52 DF,  p-value: 1.774e-09
```

Aggiungendo anche i termini di secondo ordine la bontà di adattamento migliora sensibilmente, come evidenziato dal valore di R^2 . La superficie di risposta quadratica sembra più indicata per rappresentare il trend spaziale.

3.3 Rappresentare graficamente i residui della superficie di risposta quadratica

```
data$residui<-residuals(quadr.fit)
bubble(data, "residui", col = c(mycolr, mycolb),sp.layout=list(13,11,12),do.sqrt=F)
```



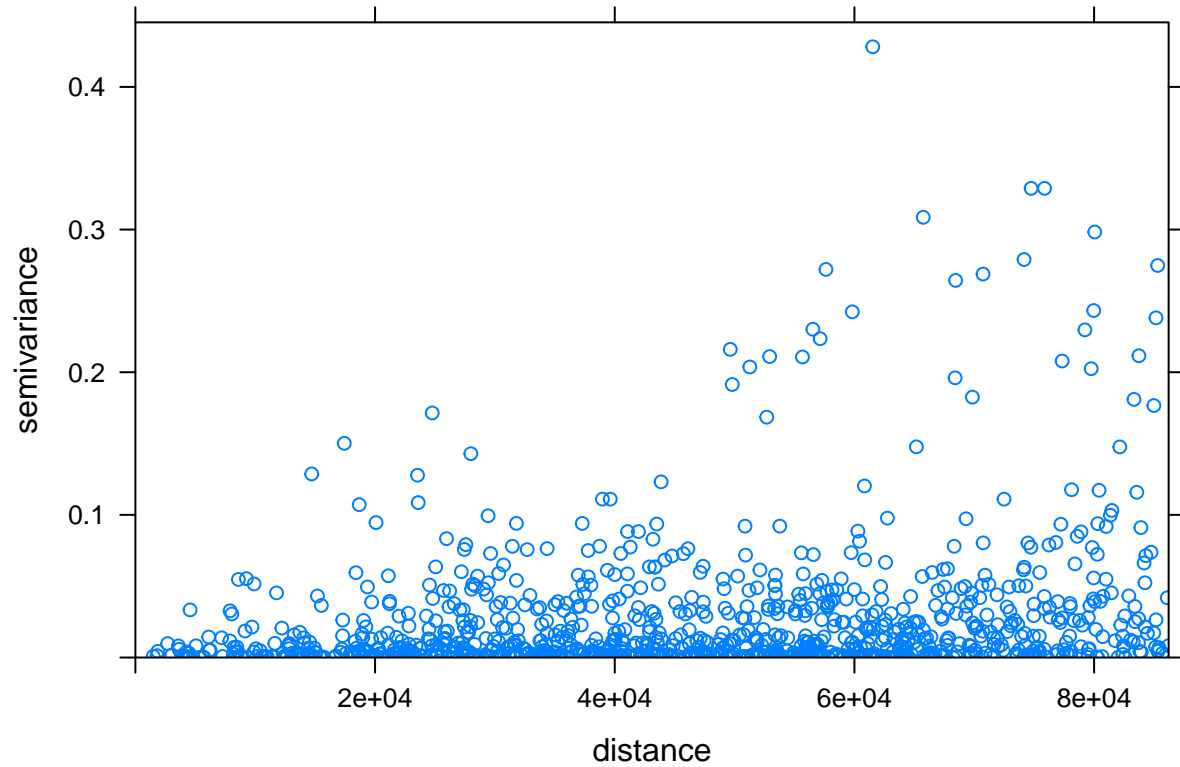
Non è facile giudicare la struttura di autocorrelazione spaziale dalla semplice osservazione del diagramma a bolle, si può comunque notare che residui di previsione positivi e negativi tendono in generale ad addensarsi in zone specifiche.

4. Il variogramma

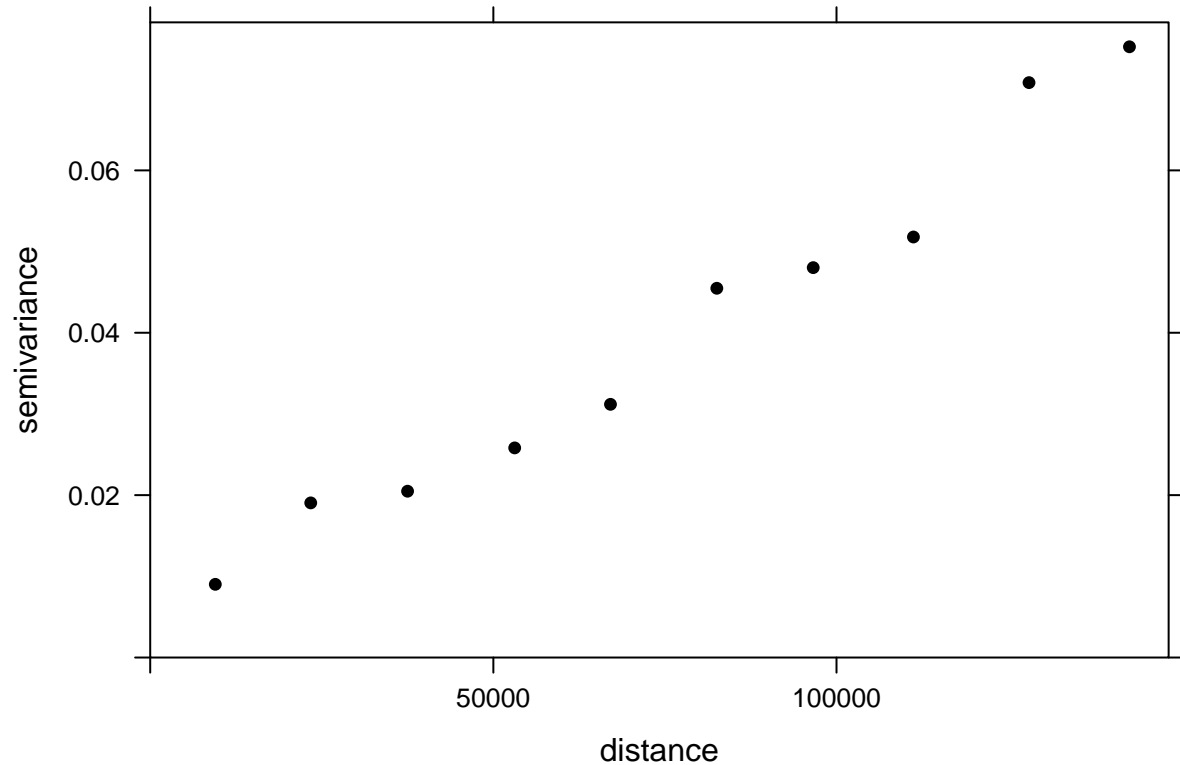
Prima di procedere all'interpolazione è necessario studiare la struttura di autocorrelazione dei dati mediante il variogramma.

4.1 Costuire il variogramma a nuvola e il variogramma empirico dei dati osservati.

```
vgm.cloud <- variogram(logpm10~1, data, cloud=T)
plot(vgm.cloud)
```



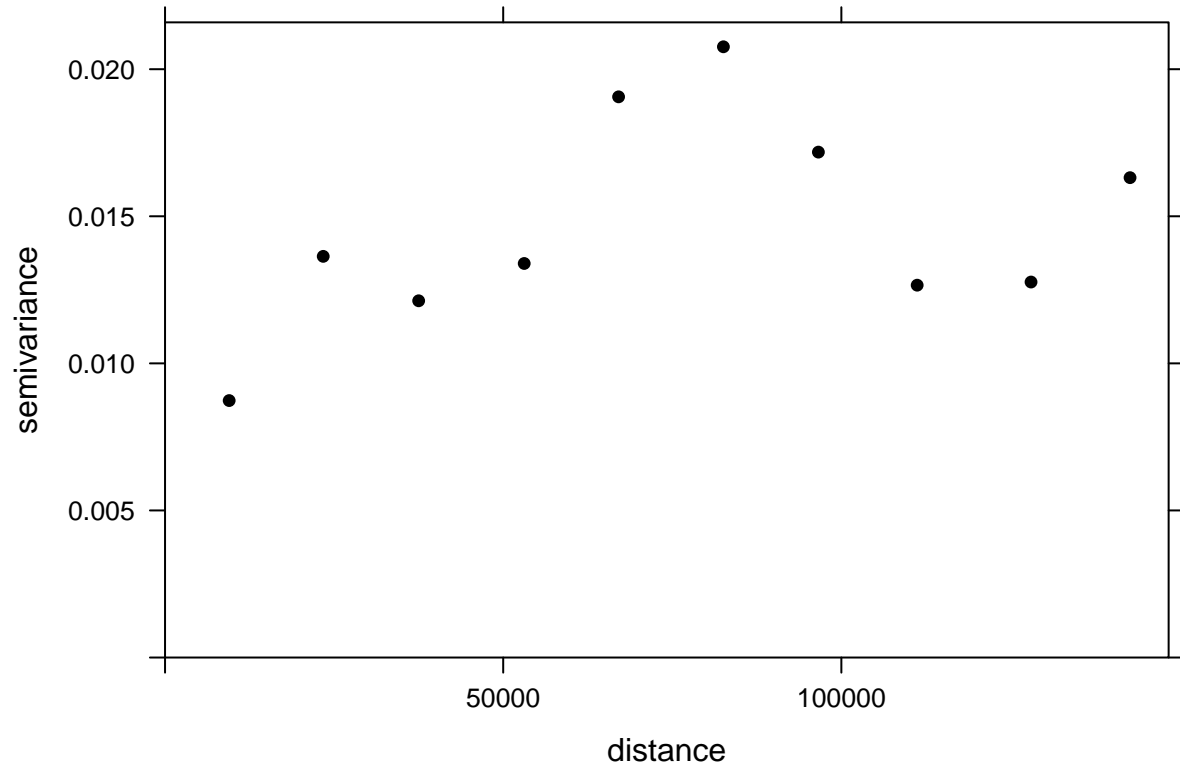
```
vgm.emp <- variogram(logpm10~1, data, cutoff=1.5e5, width=15e3)  
plot(vgm.emp, col=1, pch=16)
```

Il variogramma empirico evidenzia chiaramente che il processo osservato non è stazionario ed occorre rimuovere il trend spaziale.

4.2 Costuire il variogramma empirico dei residui della superficie di risposta quadratica.

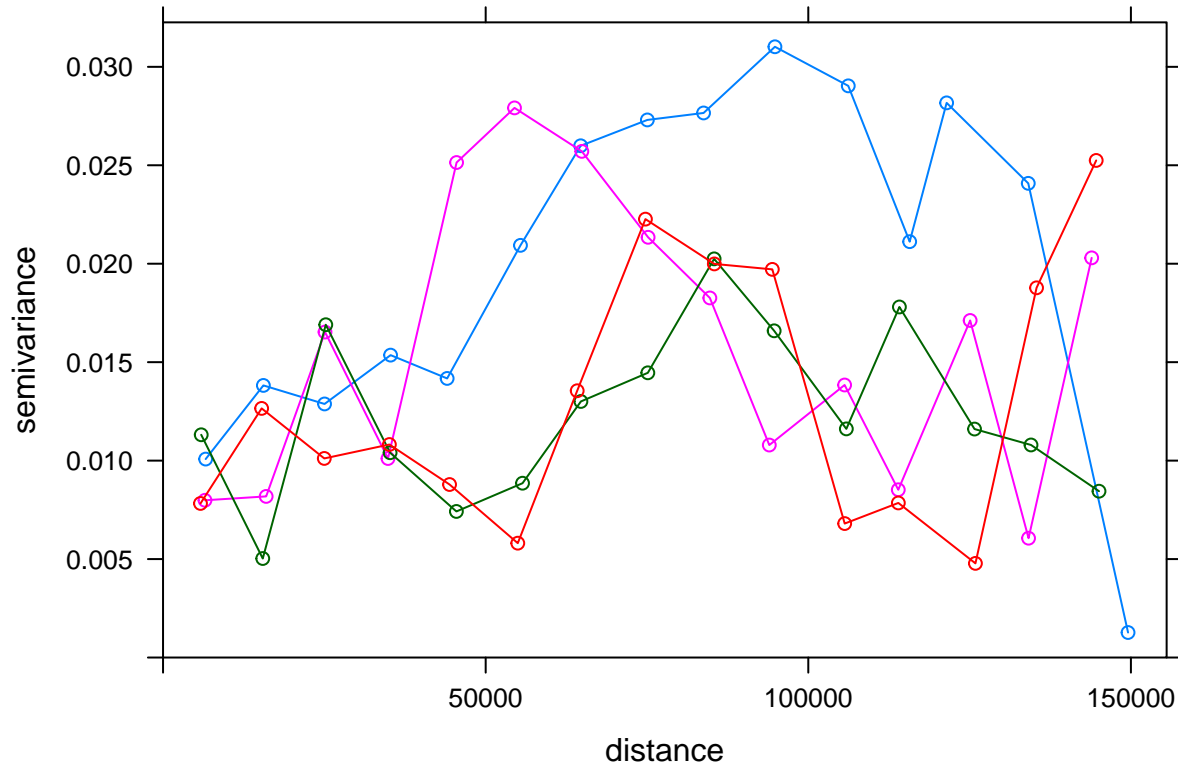
```
vgm.emp.quad <- variogram(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,cutoff=1.5e5,width=15e3)
plot(vgm.emp.quad,col=1,pch=16)
```



Dopo la rimozione del trend quadratico il variogramma empirico si stabilizza. E' bene controllare anche i variogrammi direzionali per valutare l'eventuale presenza di anisotropia.

4.3 Costruire i variogrammi direzionali.

```
vgm.dir<-variogram(logpm10~x+y+I(x^2)+I(y^2)+x*y, data, alpha=c(0,45,90,135),cutoff=1.5e5)
plot(vgm.dir,multipanel=F)
```



I variogrammi direzionali non evidenziano particolari differenze.

4.4 Stimare i variogrammi parametrici sulla base del variogramma empirico costruito in precedenza.

Per poter successivamente utilizzare il *kriging* per l'interpolazione dei dati è necessario stimare uno o più variogrammi parametrici. Utilizzando più di un modello è possibile valutare a posteriori quale fornisce i migliori risultati in termini predittivi.

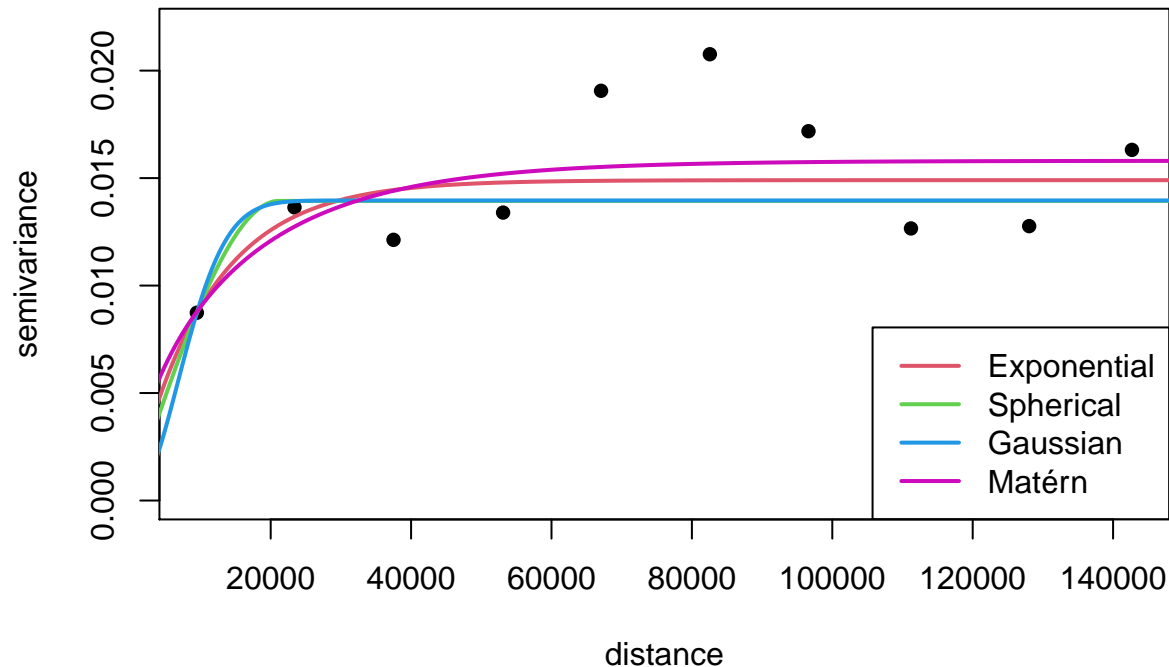
```
assex<-seq(1,to=1.5e5,by=10)
vario.expo.wls<-fit.variogram(vgm.emp.quad,model=vgm(0.1,"Exp",100))
```

```
## Warning in fit.variogram(vgm.emp.quad, model = vgm(0.1, "Exp", 100)): No
## convergence after 200 iterations: try different initial values?
```

L'algoritmo non converge perché i valori iniziali utilizzati sono troppo lontani da quelli reali.

```
assex<-seq(1,to=1.5e5,by=10)
vario.expo.wls<-fit.variogram(vgm.emp.quad,model=vgm(0.1,"Exp",1000))
y.expo<-variogramLine(vario.expo.wls,dist_vector=assex)
vario.sphe.wls<-fit.variogram(vgm.emp.quad,model=vgm(0.1,"Sph",100000))
y.sphe<-variogramLine(vario.sphe.wls,dist_vector=assex)
vario.gaus.wls<-fit.variogram(vgm.emp.quad,model=vgm(0.1,"Gau",10000))
y.gaus<-variogramLine(vario.gaus.wls,dist_vector=assex)
vario.mate.wls<-fit.variogram(vgm.emp.quad,model=vgm(0.1,"Mat",1000),fit.kappa=T)
y.mate<-variogramLine(vario.mate.wls,dist_vector=assex)
plot(vgm.emp.quad$dist,vgm.emp.quad$gamma,xlab="distance",ylab="semivariance",pch=16,ylim=c(0,0.022))
lines(y.expo,col=2,lwd=2)
lines(y.sphe,col=3,lwd=2)
```

```
lines(y.gaus,col=4,lwd=2)
lines(y.mate,col=6,lwd=2)
legend("bottomright",legend=c("Exponential","Spherical","Gaussian","Matérn"),col=c(2:4,6),lwd=2)
```



Il variogramma gaussiano ed il variogramma sferico sono praticamente identici e anche il variogramma esponenziale e il variogramma di Matérn sono molto simili. Utilizzeremo questi variogrammi per prevedere i livelli di PM10 su tutta la Lombardia.

5. La previsione

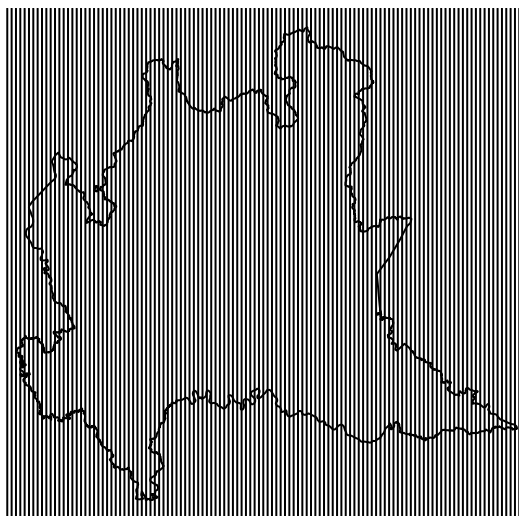
Il variogramma parametrico è necessario per poter poi interpolare i valori osservati sulla regione di studio mediante il *kriging*. A questo fine occorre in primo luogo individuare i punti nei quali effettuare la previsione. Solitamente si utilizzano i nodi di una griglia regolare.

5.1 Costruire una griglia di punti per la previsione

Mentre nel caso di una regione di studio di forma regolare (tipicamente rettangolare o quadrata) la definizione dei punti nei quali effettuare la previsione è relativamente semplice, le cose diventano più complesse nel caso di una regione di studio di forma irregolare.

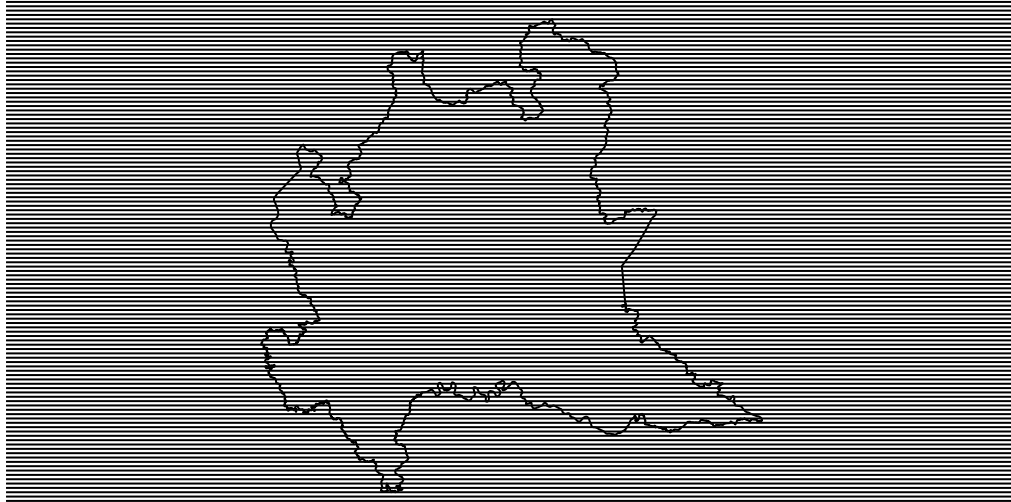
In primo luogo occorre individuare le ascisse e le ordinate dei punti che costituiscono la griglia. Per fare questo è necessario decidere il passo da utilizzare, che può essere uguale per ascisse ed ordinate (griglia a maglia quadrata) o diverso (griglia a maglia rettangolare).

```
passo<-2000
punti.x<-seq(from=bbbox(lombardia)[1,1]-4000,length=120,by=passo)
plot(lombardia)
abline(v=punti.x)
```



La funzione `bbox` estrae il valore minimo ed il valore massimo di longitudine e latitudine dall'oggetto di classe `sp` sul quale è applicata. La funzione `seq` genera una sequenza di valori a partire da quello iniziale (`from=`), con incrementi costanti e pari al valore specificato all'argomento `by`. La lunghezza della sequenza è individuata dall'argomento `length`. Per il momento abbiamo definito le ascisse dei punti che costituiscono la griglia. Allo stesso modo possiamo definire le ordinate.

```
punti.y<-seq(from=bbox(lombardia)[2,1]-4000,by=passo,length=120)
plot(lombardia)
abline(h=punti.y)
```



Una volta individuate le ascisse e le ordinate si posso identificare tutte le coppie possibili di coordinate mediante la funzione `expand.grid` ed il data.frame che contiene i punti può essere definito di tipo spaziale.

```
cp<-expand.grid(x=punti.x,y=punti.y)
coordinates(cp)<- ~ x+y
gridded(cp)<-T
proj4string(cp)<-proj4string(lombardia)
```

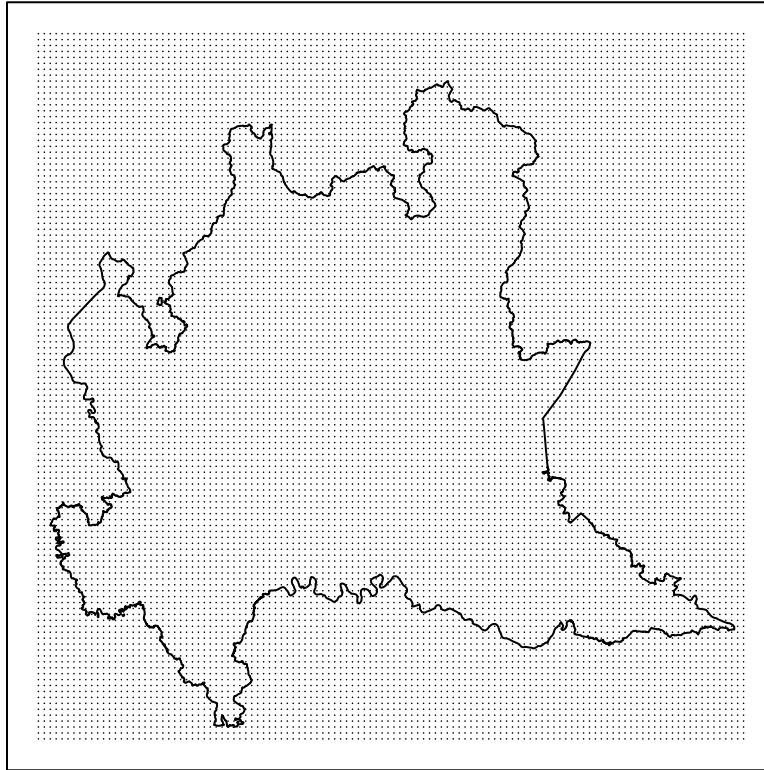
```
## Warning in proj4string(lombardia): CRS object has comment, which is lost in output; in tests, see
## https://cran.r-project.org/web/packages/sp/vignettes/CRS_warnings.html
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj
## = prefer_proj): Discarded datum Unknown based on International 1909 (Hayford)
## ellipsoid in Proj4 definition
```

```
summary(cp)
```

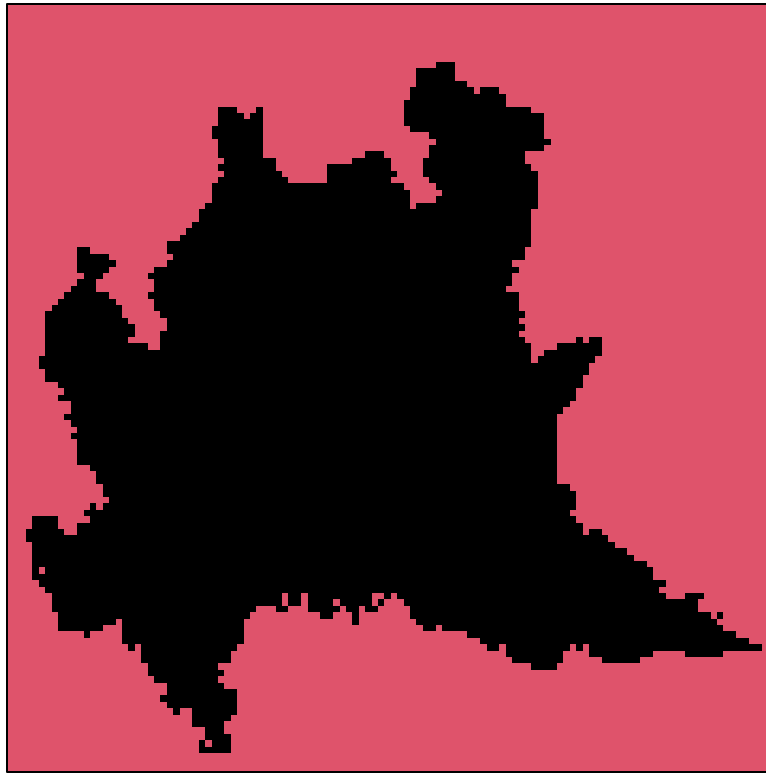
```
## Object of class SpatialPixels
## Coordinates:
##           min           max
## x  455707.3  695707.3
## y  4942603.0  5182603.0
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=32 +ellps=intl +units=m +no_defs]
## Number of points: 14400
## Grid attributes:
##   cellcentre.offset cellsize cells.dim
```

```
## x      456707.3    2000    120
## y      4943603.0    2000    120
spplot(cp,sp.layout=list(13),pch=16,col.regions=1,cex=0.1,colorkey=list())
```

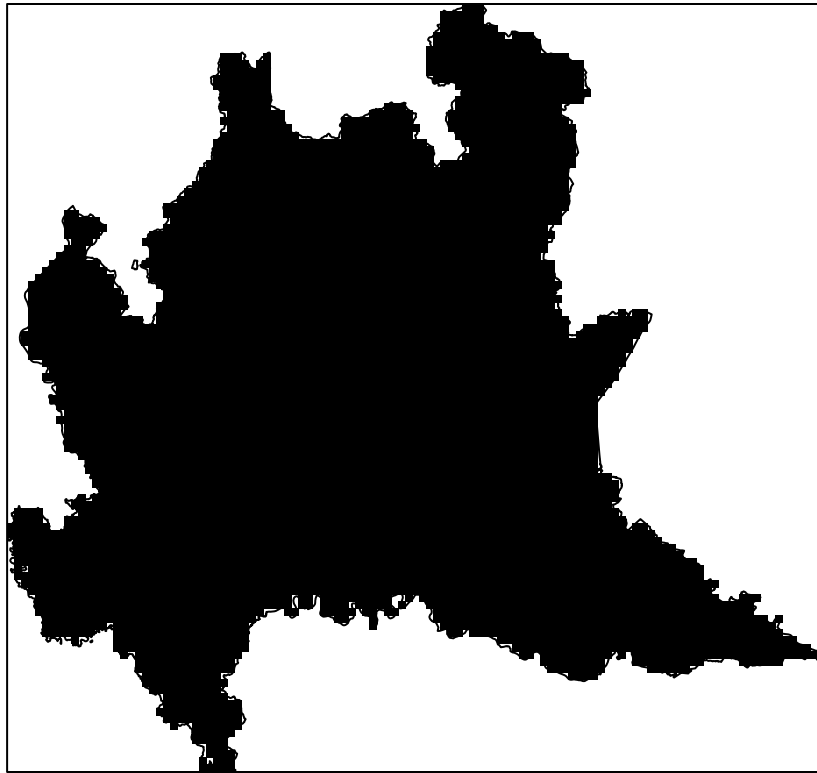


Poiché abbiamo creato una griglia regolare che abbiamo sovrapposto alla regione di studio, molti punti cadono al di fuori dei confini regionali. Utilizzando la funzione `over` è possibile individuare i punti all'interno e all'esterno dei confini regionali e successivamente scegliere solo quelli interni.

```
a<-over(cp,lombardia)
cp$indi<-!is.na(a$COD_REG)
spplot(cp,col.regions=2:1,pch=16,cex=0.1,sp.layout=list(13),colorkey=F)
```



```
cp<-cp[cp$indi,]  
spplot(cp,pch=16,cex=0.1,sp.layout=list(13),col.regions=1,colorkey=F)
```

5.2 Interpolare i livelli di PM10 utilizzando l'*inverse distance weighting*

L'*inverse distance weighting* definisce i pesi della combinazione lineare in base al reciproco della distanza elevata al coefficiente p . E' possibile provare valori di p diversi in modo da evidenziare l'effetto di questo coefficiente.

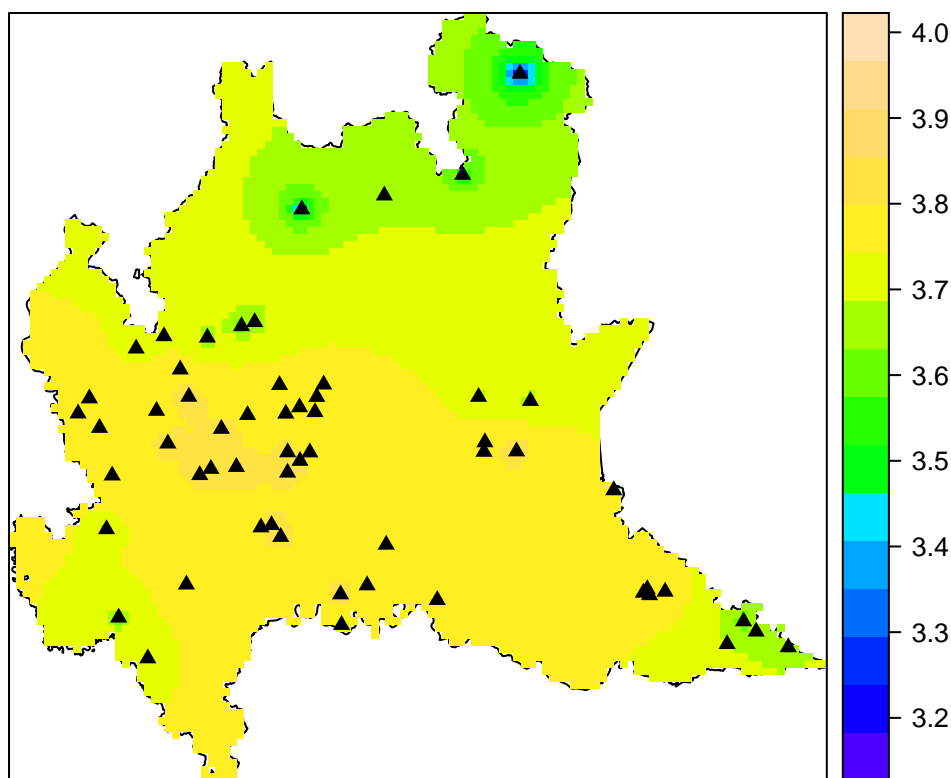
```
proj4string(data)<-proj4string(lombardia)
```

```
## Warning in proj4string(lombardia): CRS object has comment, which is lost in output; in tests, see  
## https://cran.r-project.org/web/packages/sp/vignettes/CRS_warnings.html
```

```
idw.est.1 <- idw(formula=logpm10~1,data, newdata=cp,idp=1)
```

```
## [inverse distance weighted interpolation]
```

```
l4<-list(data,pch=17,col=1)  
spplot(idw.est.1,zcol="var1.pred",sp.layout=list(l3,l4),  
       col.regions=topo.colors(100))
```

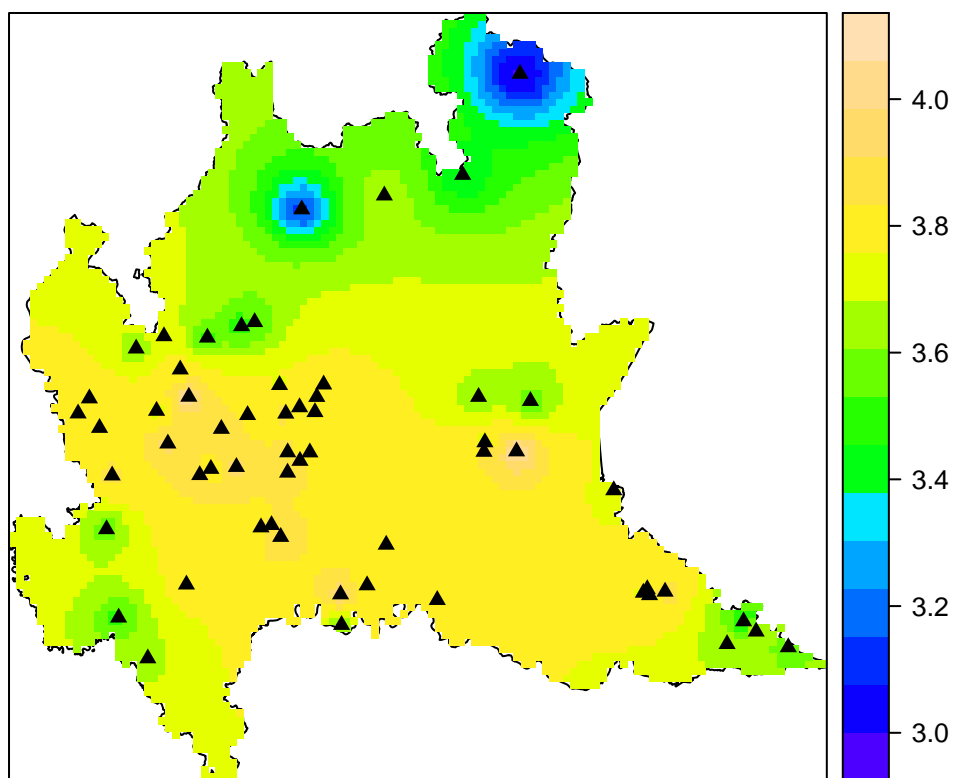


Larga parte della regione è caratterizzata da livelli medio alti di PM10, in particolare nell'area milanese (zone in diverse tonalità di giallo). Si notano in diverse zone della mappa gli artefatti tipici dell'*inverse distance weighting*.

```
idw.est.2 <- idw(formula=logpm10~1,data, newdata=cp,idp=2)
```

```
## [inverse distance weighted interpolation]
```

```
spplot(idw.est.2,zcol="var1.pred",sp.layout=list(13,14),
       col.regions=topo.colors(100))
```

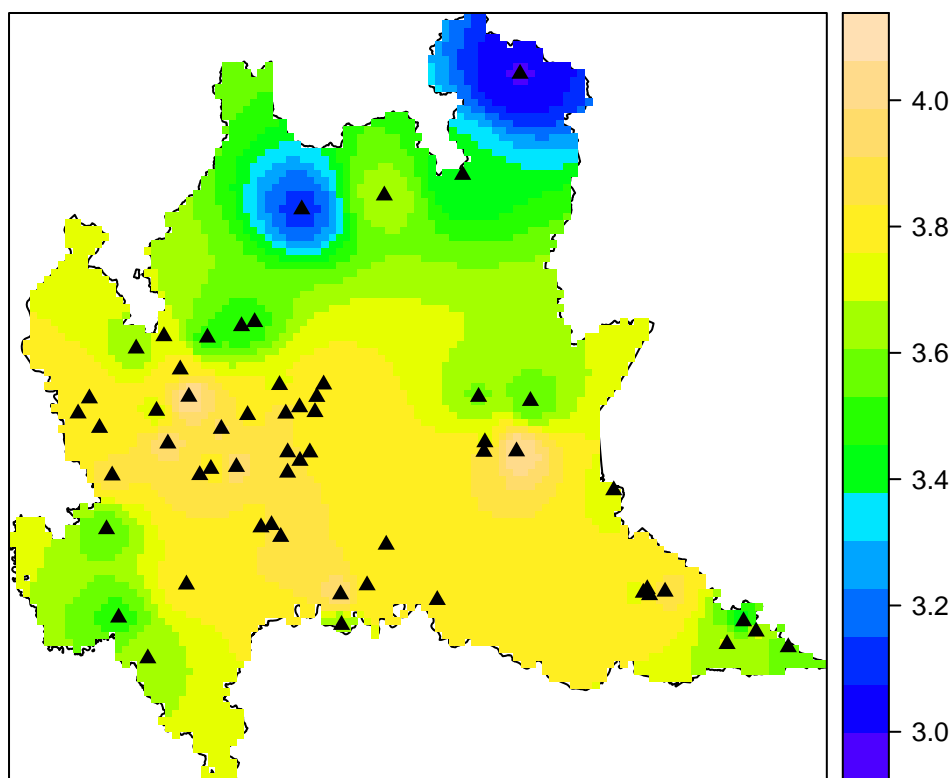


Aumentando il valore di p i pesi decrescono più velocemente al crescere della distanza. Per questo motivo gli artefatti diventano più marcati.

```
idw.est.3 <- idw(formula=logpm10~1,data, newdata=cp,idp=3)
```

```
## [inverse distance weighted interpolation]
```

```
spplot(idw.est.3,zcol="var1.pred",sp.layout=list(13,14),
       col.regions=topo.colors(100))
```



L'effetto è ancora più evidente con $p=3$.

6. Il *kriging*

Il *kriging* determina i pesi in funzione della autocorrelazione spaziale che caratterizza il processo studiato, che viene determinata sulla base del variogramma parametrico scelto per la rappresentazione. Poiché è possibile utilizzare diverse forme funzionali è opportuno adottare un criterio per scegliere. Una possibilità è data dall'indice di validazione incrociata, che approssima l'errore quadratico medio di previsione. E' possibile quindi calcolare il coefficiente CV per i diversi modelli precedentemente adattati e poi scegliere quello cui corrisponde il valore minore.

6.1 Utilizzare la validazione incrociata per determinare il variogramma parametrico che minimizza l'errore quadratico medio di previsione.

```
krig.cv.exp <- krige.cv(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,
                        nfold=dim(data)[1], model = vario.expo.wls)
CV.exp<-mean(krig.cv.exp$residual^2)
CV.exp
```

```
## [1] 0.01851021
```

La funzione `krige.cv` costruisce un oggetto di classe `SpatialPointsDataFrame` che contiene, i valori previsti mediante validazione incrociata, la varianza dei valori previsti, i valori osservati, i residui (differenza tra valori osservati e attesi) e i residui standardizzati. L'indice di validazione incrociata è la media dei residui elevati al quadrato.

```
krig.cv.sph <- krige.cv(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,
                        nfold=dim(data)[1], model = vario.sphe.wls)
```

```
CV.sph<-mean(krig.cv.sph$residual^2)
CV.sph
```

```
## [1] 0.01870499
```

```
krig.cv.gau <- krige.cv(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,
                        nfold=dim(data)[1], model = vario.gaus.wls)
CV.gau<-mean(krig.cv.gau$residual^2)
CV.gau
```

```
## [1] 0.02137747
```

```
krig.cv.mat <- krige.cv(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,
                        nfold=dim(data)[1], model = vario.mate.wls)
CV.mat<-mean(krig.cv.mat$residual^2)
CV.mat
```

```
## [1] 0.01817072
```

I valori dell'indice di validazione incrociata sono molto simili tra loro (si discosta in misura migliore solo quello relativo al variogramma gaussiano). Il variogramma di Matérn è quello che fornisce il valore minimo dell'indice e quindi quello che utilizzeremo per la previsione basata sul *kriging*.

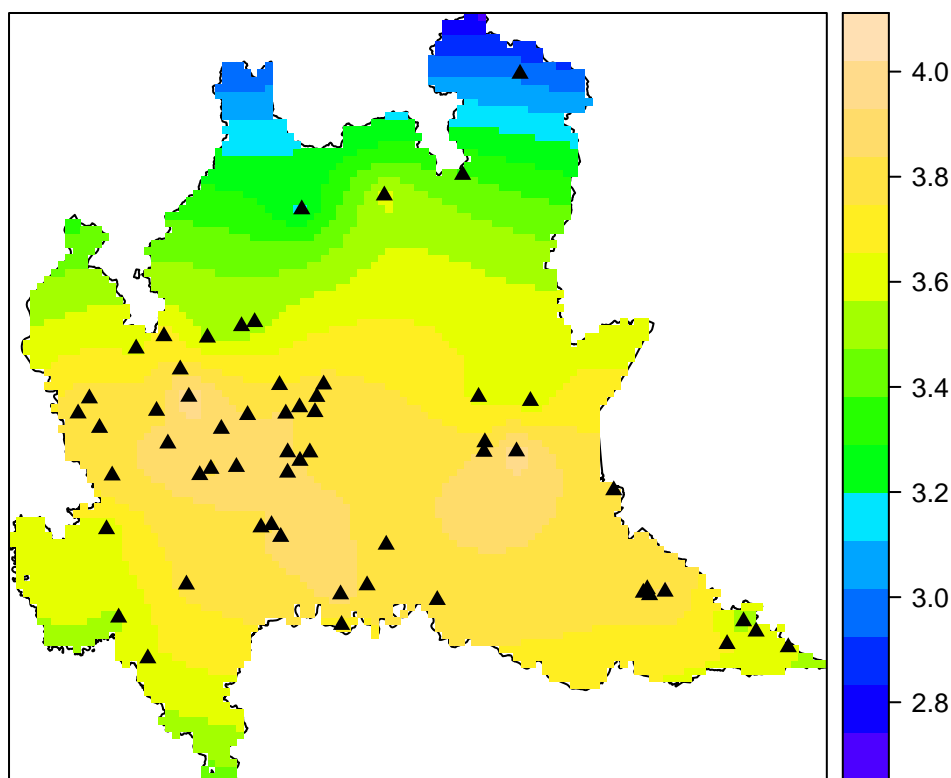
6.2 Utilizzare il *kriging* con il variogramma di Matérn per calcolare la previsione dei livelli di PM10 nella regione di studio.

La funzione `krige` consente di costruire le previsioni in corrispondenza dei nodi della griglia definita in precedenza.

```
krig.mat <- krige(logpm10~x+y+I(x^2)+I(y^2)+x*y, data,
                  newdata=cp, model = vario.mate.wls)
```

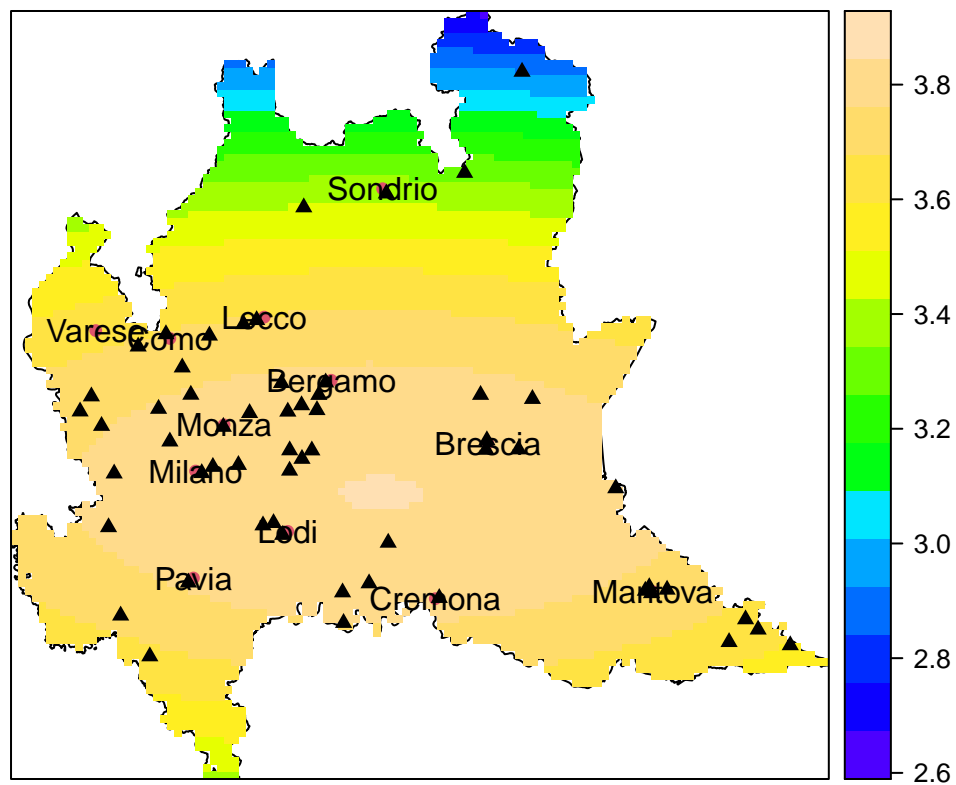
```
## [using universal kriging]
```

```
spplot(krig.mat,zcol="var1.pred",sp.layout=list(l3,l4),
        col.regions=topo.colors(100))
```



La mappa descrive la distribuzione spaziale del PM10 come stimata sulla base delle osservazioni relative alle centraline. Si notano due aree a livello molto elevato, la prima tra le province di Milano, Monza e Lodi, la seconda in prossimità di Brescia. A titolo di confronto è possibile costruire anche la mappa basata sulla sola componente di grande scala.

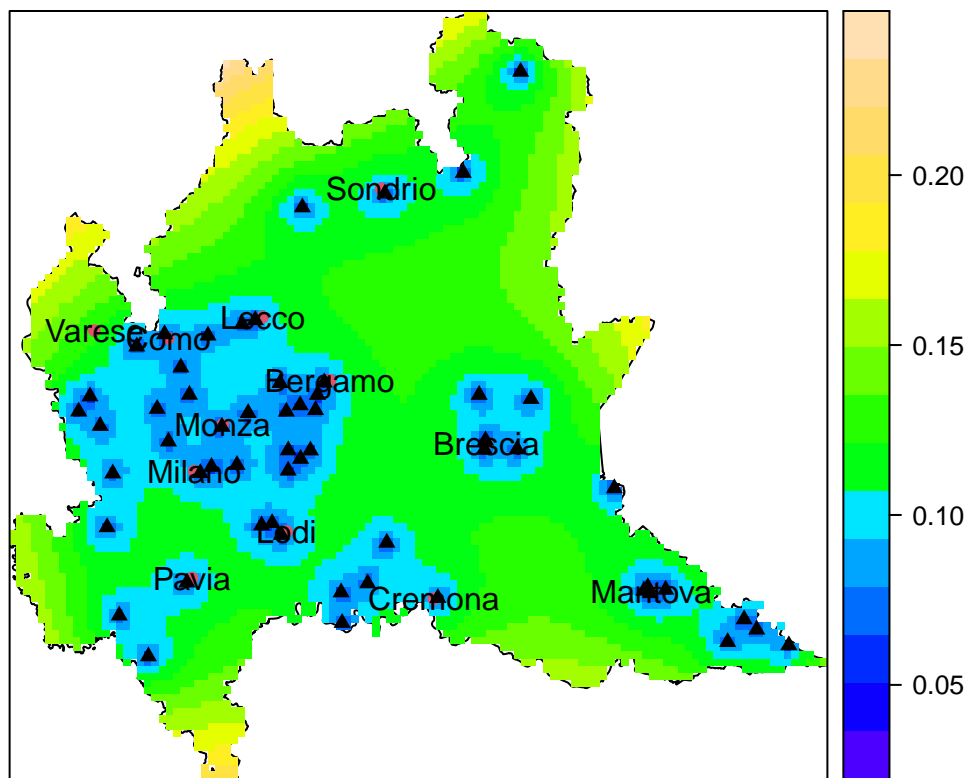
```
cp$grandescala<-predict(quadr.fit,newdata=cp)
spplot(cp,zcol="grandescala",sp.layout=list(11,12,13,14),
       col.regions=topo.colors(100))
```



Confrontando le due mappe si notano differenze abbastanza importanti, pur rimanendo inalterata la tendenza di fondo. Il confronto aiuta a chiarire la differenza tra la variabilità a grande scala e quella a piccola scala.

Oltre alla previsione *kriging* è possibile rappresentare anche l'errore standard della previsione, sempre calcolato dalla funzione *krige*.

```
krig.mat$var1.sterr<-sqrt(krig.mat$var1.var)
spplot(krig.mat,zcol="var1.sterr",sp.layout=list(l1,l2,l3,l4),
        col.regions=topo.colors(100))
```

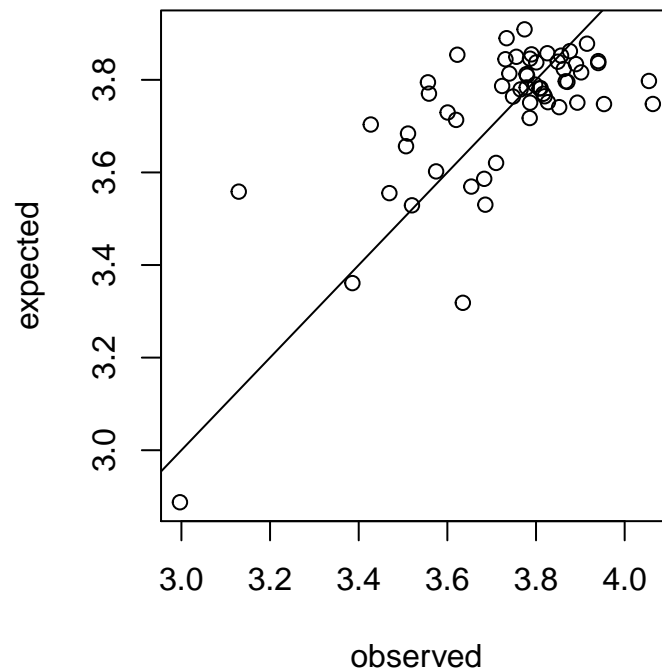


Si nota che in prossimità delle centraline l'errore di previsione è basso e cresce allontanandosi da queste formando una caratteristica mappa a macchie tondeggianti. Questo risultato non è un artefatto del metodo, ma semplicemente il riflesso della quantità di informazione sulla quale sono basate le previsioni, che è elevata in prossimità dei punti di osservazione e decresce allontanandosi da questi.

6.3 Utilizzare la validazione incrociata per individuare eventuali problemi sulla previsione relativa alle singole centraline.

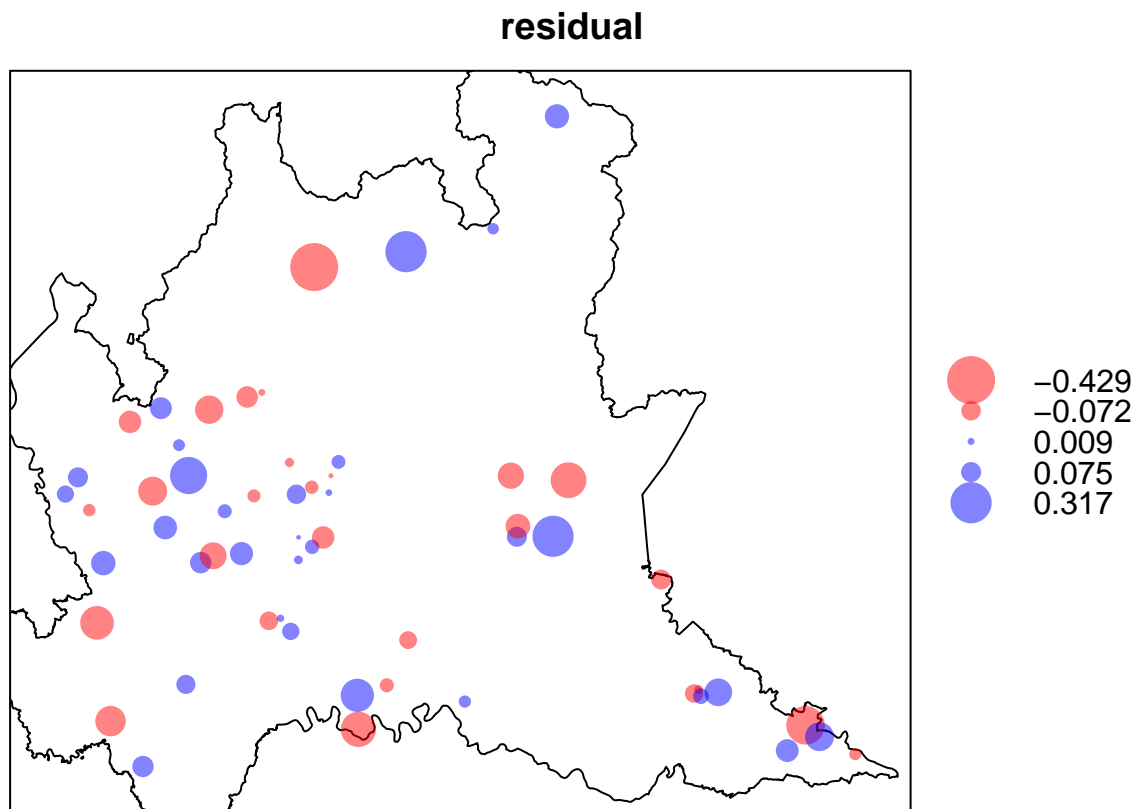
La validazione incrociata, oltre a consentirci di calcolare l'indice di validazione incrociata, che è aggregato su tutte le osservazioni, consente anche di valutare la bontà della previsione in corrispondenza di ogni singola osservazione. In primo luogo si possono confrontare i valori osservati con quelli predetti mediante un diagramma di dispersione.

```
par(pty="s")
plot(krig.cv.mat$observed, krig.cv.mat$observed - krig.cv.mat$residual,
     xlab="observed", ylab="expected")
abline(a=0, b=1)
```

La maggior parte dei punti si addensa nella parte alta del grafico perché la maggior parte delle centraline presenta valori elevati di PM10. Gli scostamenti tra valori osservati e valori previsti non sono del tutto trascurabili.

```
bubble(krig.cv.mat, "residual", col = c(mycolr, mycolb), sp.layout=list(13))
```



Non è facile giudicare la struttura di autocorrelazione spaziale dalla semplice osservazione del diagramma a bolle, si può comunque notare che residui di previsione positivi e negativi non tendono in generale ad addensarsi in zone specifiche.