Geostatistics project:

In this project I do data transformation, visualization and analysis of the gas initial potential and cumulative production at different locations from the geostatistics perspectives.

Analysis:

From the covariograms we can see the covariance of points which are close are bigger than the covariance of points which are far away. The size of covariances decreases with increase of distances, this feature fits with the general trend of covariogram plots. After excluding the extreme values, I plot plots of the data points. I notice the density of initial potential points between the range of 0-500 is much higher, and the density of cumulative production points is higher in the range of 0-500.

For the initial potential data points, I plot the semivariograms of four different directions and find the dir=0 fit the points best. Then I use the different weights, n pair , cressies, and equal weights to fit the exponential model. The plots looks similar so there is no big difference between the use of weights for these data points. Comparing the exponential model with the spherical model I use the cross validation method. I find the PRESS of the exponential model is smaller than the PRESS of spherical model. So for the initial potential data points it is better to use the exponential model. Then I draw the plots of predict values and variances using ordinary kriging and universal kriging and get similar results.

Then I do similar things on the cumulative production data points. The semivariogram with dir=pi/4 seems to fit the points best. Then comparing the model with different weights the one with the npair seems to fit the points better but there is no big difference between the models with different weights. Then I draw the plots of predict values and variances using ordinary kriging and universal kriging and get similar results. Then comparing the difference of the PRESS of exponential model and spherical model, I notice the PRESS of the spherical model is smaller. Thus it may be better to use the spherical model for cumulative data points.

From the data distribution and bubble plots I notice there are lots of points in the cumulative production that is -999. It does not make sense for a production to be negative. I search online and find it may be because that place stops production. So I redo the analysis on the cumulate production points without points with negative production and have some new plots. It seems that dir=0 fits the points best, and for the PRESS for exponential and spherical model the difference is too small that I think I can conclude there is no big difference in using the two model base on the results of cross validation.

173project

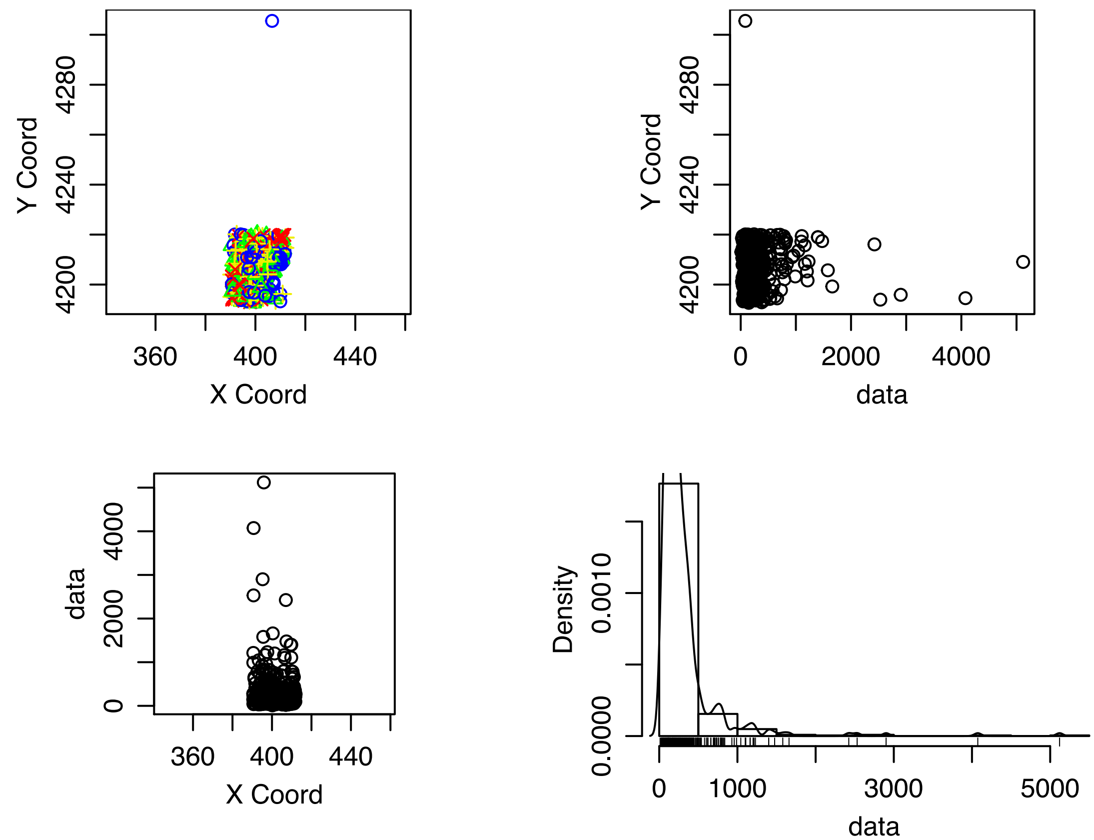
Wenxin Zhou

3/19/2019

```
library(geoR)
```

```
## --------------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2.1 (built on 2016-05-02) is now loaded
## --------------------------------------------------------------
```

```
project<-read.table("/Users/sophiazhou/Downloads/Gas Initial Potential and Cumulative Production (1).tx
a<-as.geodata(project)
plot(a)
```




```
options(warn=-1)

#We can see there are many outliers so I remove the outliers and plot a new graph
s1=summary(project$easting.Km.)
s2=summary(project$northing.Km.)
s3=summary(project$Initial_Potential)
s4=summary(project$Cumulative_Production)
s1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   390.5   396.3   400.6   401.0   405.7   412.0
```

```
s2
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4193    4201    4209    4208    4214    4306
```

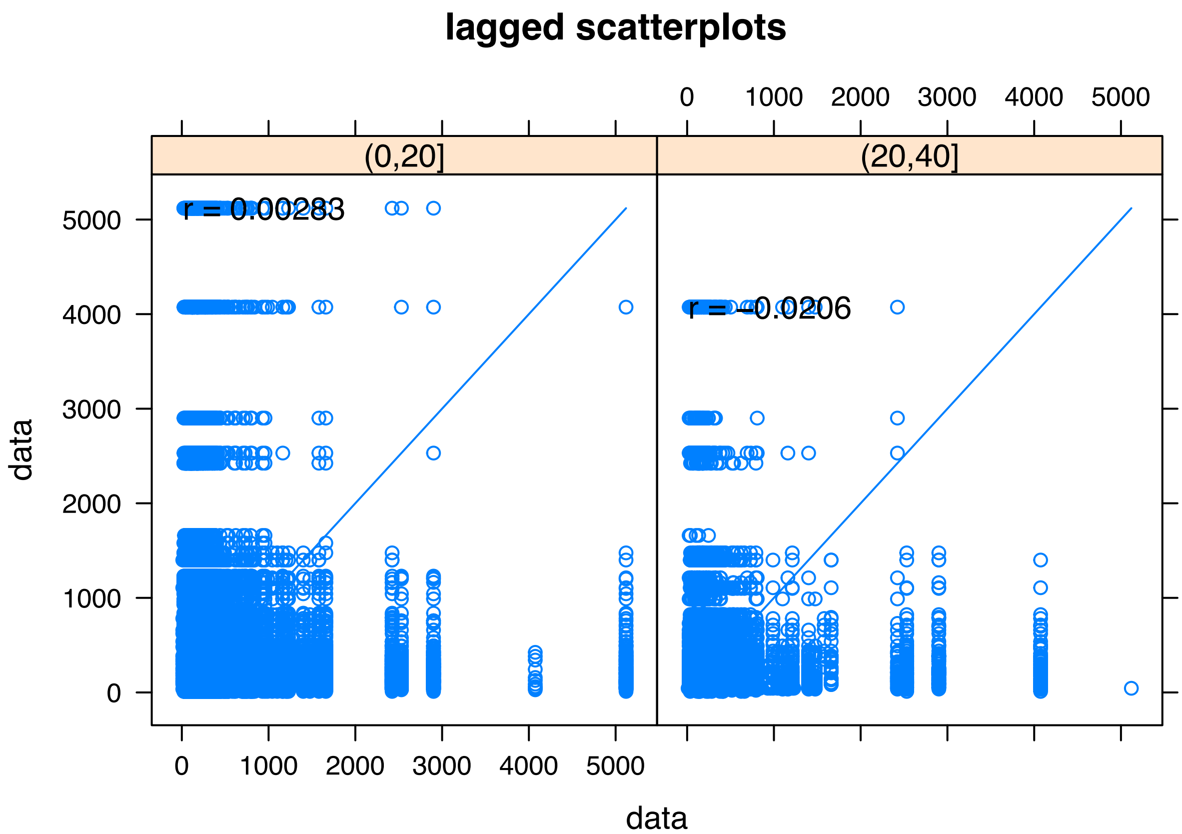
```
s3
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    11.0   119.0   199.0   305.3   335.5  5120.0
```

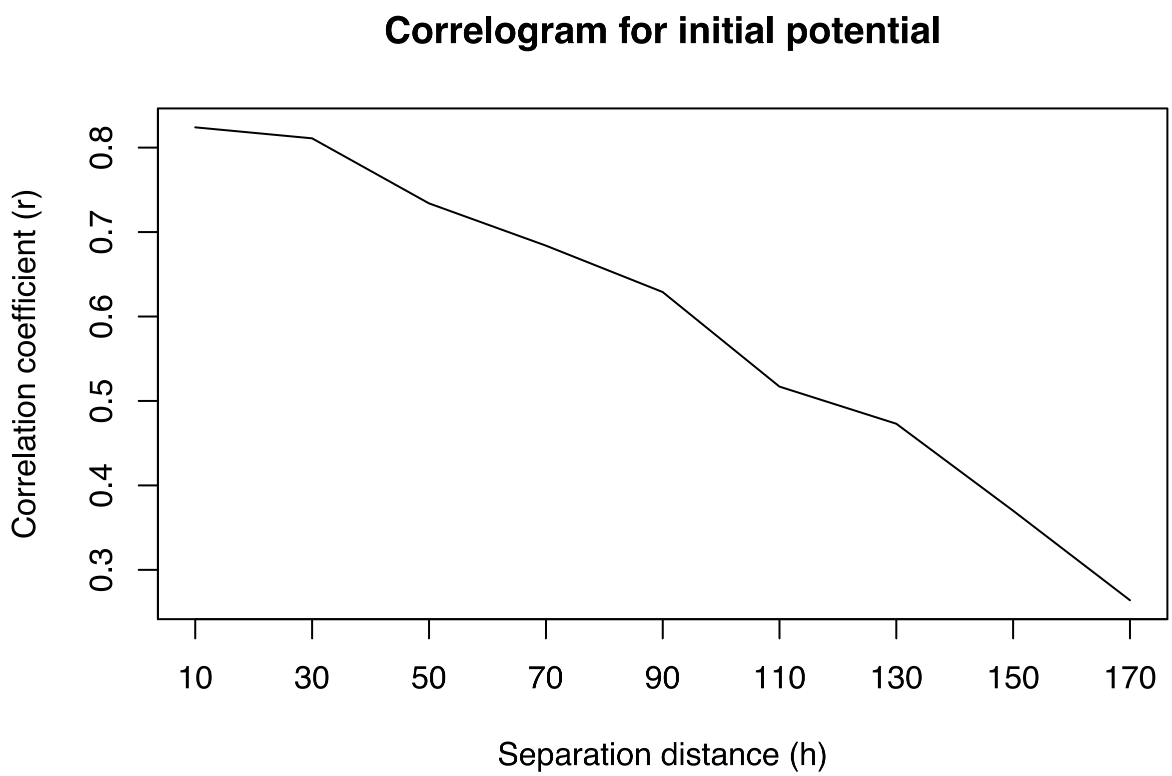
```
s4
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -999.00   35.00   95.50  -31.14  183.25 1574.00
```

```
extreme=project$northing.Km.>4300
o1=project$Initial_Potential>s3[[5]]+(s3[[5]]-s3[[2]])*1.5
o2=project$Initial_Potential<s3[[2]]-(s3[[5]]-s3[[2]])*1.5
notoutlier=project$Initial_Potential[!c(o1,o2)]
b<-cbind(project$easting.Km.[!extreme],project$northing.Km.[!extreme],project$Initial_Potential[!extreme

##create h-scatterplots
c<-as.data.frame(b)
names(c) <- c("x", "y", "data")
library(gstat)
library(sp)
coordinates(c) <- ~x+y

qq <- hscat(data~1, c, c(0,20,40,60,80,100,120,140,160,180))

plot(qq, main="h-scatterplots")
```



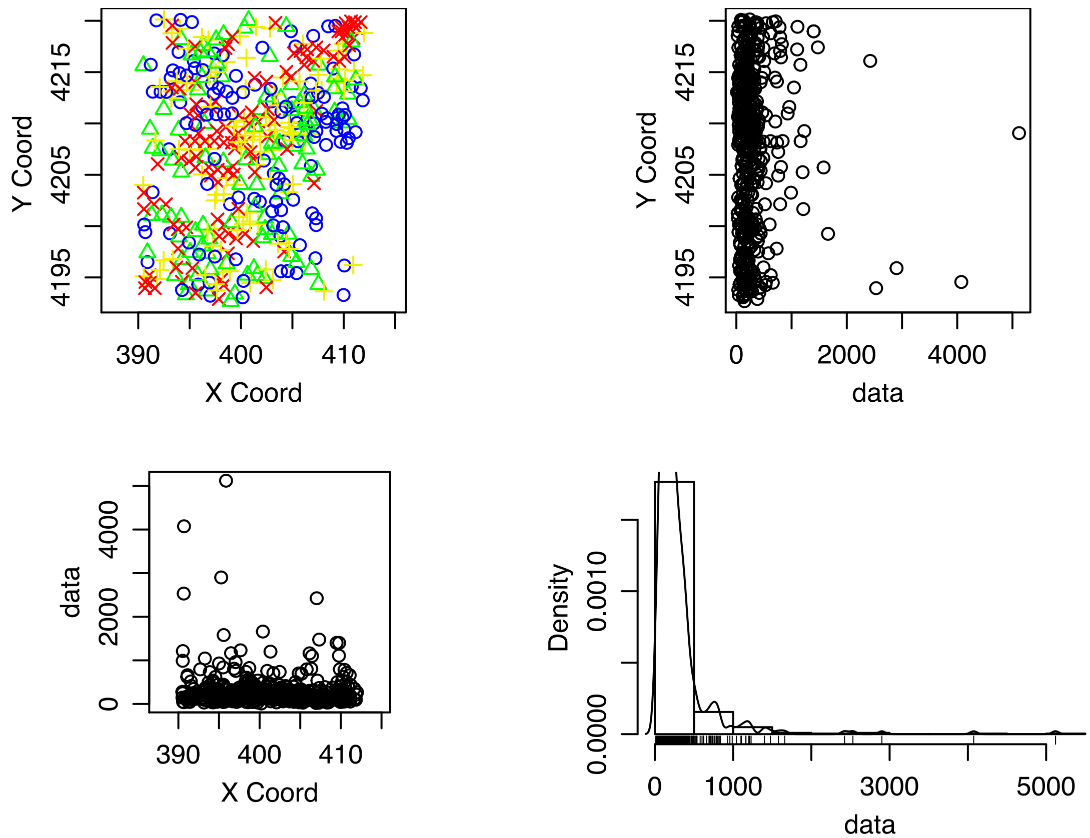

```r
#Correlogram for initial potential
plot(c(10,30,50,70,90,110,130,150,170), c(0.824, 0.811,0.734,0.684,0.629,0.517,0.473,0.37, 0.264), type=


axis(1, at=seq(10, 190, by=20),labels=seq(10, 190, by=20))


axis(2, at=seq(0, 1, by=0.1),labels=seq(0, 1, by=0.1))
```




```
b<-as.geodata(b)
#The graph without outliers
summary(b)
```

```
## Number of data points: 451
##
## Coordinates summary
##       Coord1   Coord2
## min 390.489 4192.684
## max 411.986 4220.133
##
## Distance summary
##          min          max
##  0.02109502 33.34321307
##
## Data summary
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   11.0000  119.0000  200.0000  305.7583  336.0000 5120.0000
```

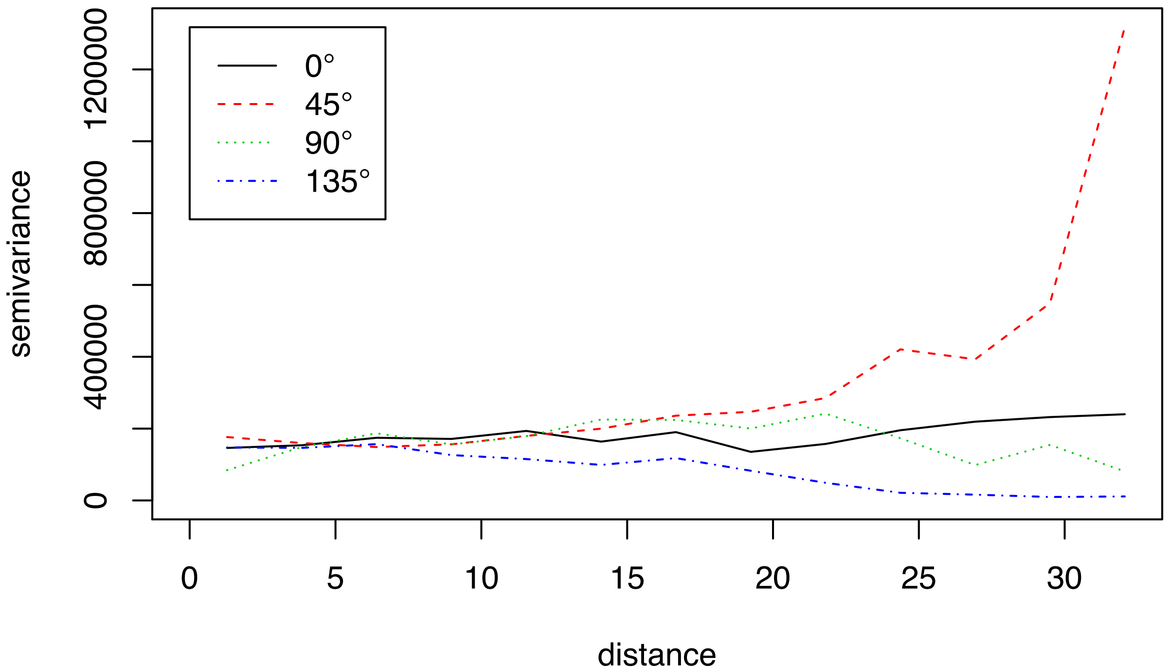
```
plot(b)
```




```r
#analize the initial portential
#plot the cariograms in for different directions
var1<-variog4(b)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 90 degrees (1.571 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 135 degrees (2.356 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing omnidirectional variogram
```

```r
plot(var1)
```




```r
#Compute the sample variogram and fit a model to it:
var1<-variog(b,dir=pi/4,estimator.type="modulus",max.dist = 24)
```

```
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
```

```r
#Plot the sample variogram:
plot(var1)

fit1<-variofit(var1,cov.model="exp",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
#Obtain the estimates of the parameters:
fit1
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
## parameter estimates:
##       tausq     sigmasq         phi
##   26649.108 8721605.593    4439.853
## Practical Range with cor=0.05 for asymptotic range: 13300.61
##
## variofit: minimised weighted sum of squares = 393942809378
```

```r
#Plot the model variogram:
lines(fit1)
initial.values<-expand.grid(seq(10,100,by=10),seq(10000,15000,by=200))
#npairs weights
fit1l<-variofit(var1,cov.model="exp",ini.cov.pars=initial.values,wei="npairs", fix.nugget=FALSE, nugget=
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
```

```
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 33316146110478.3
```

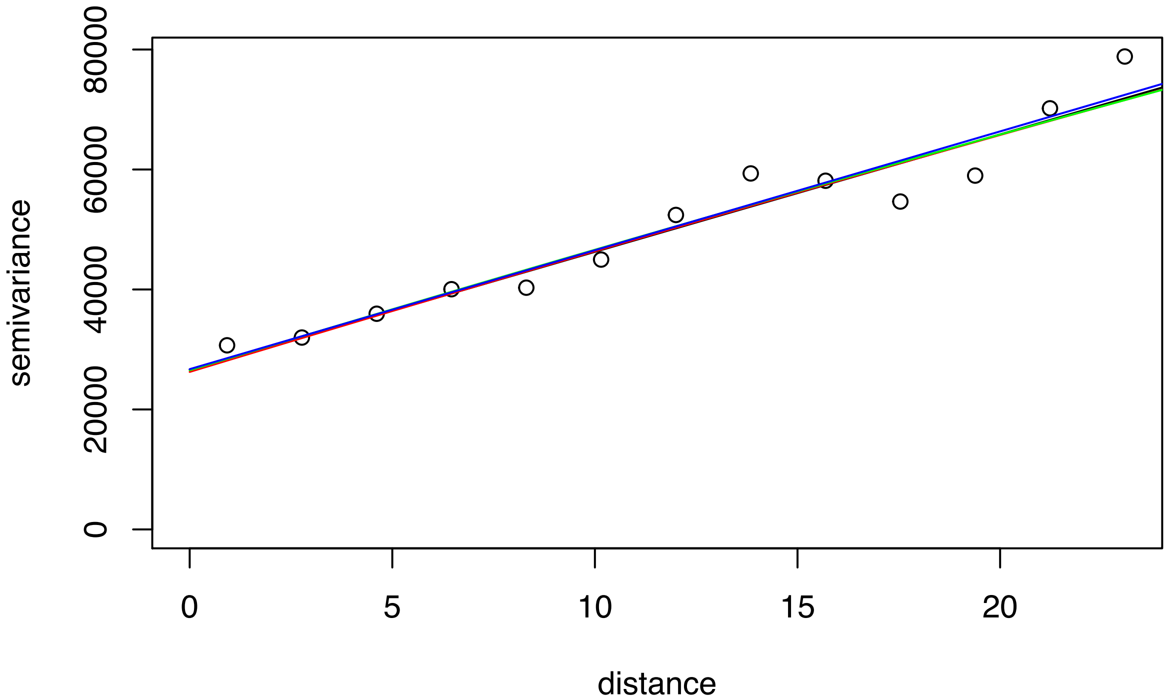
```r
lines(fit1l,lty=1,col="red")
#cressies weights
fit2l<-variofit(var1,cov.model="exp",weights="cressie",ini.cov.pars=initial.values,fix.nugget=FALSE,nugg
```

```
## variofit: covariance model used is exponential
## variofit: weights used: cressie
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 83289.012832511
```

```r
lines(fit2l,lty=1,col="green")
#equal weights (simply OLS):
fit3l<-variofit(var1,cov.model="exp", ini.cov.pars=initial.values,weights="equal",fix.nugget=FALSE,nugge
```

```
## variofit: covariance model used is exponential
## variofit: weights used: equal
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 14703706263.1898
```

```r
lines(fit3l,lty=1,col="blue")
```




```
#create grid
x.range<-as.integer(range(b[[1]][,1]))
x.range
```

```
## [1] 390 411
```

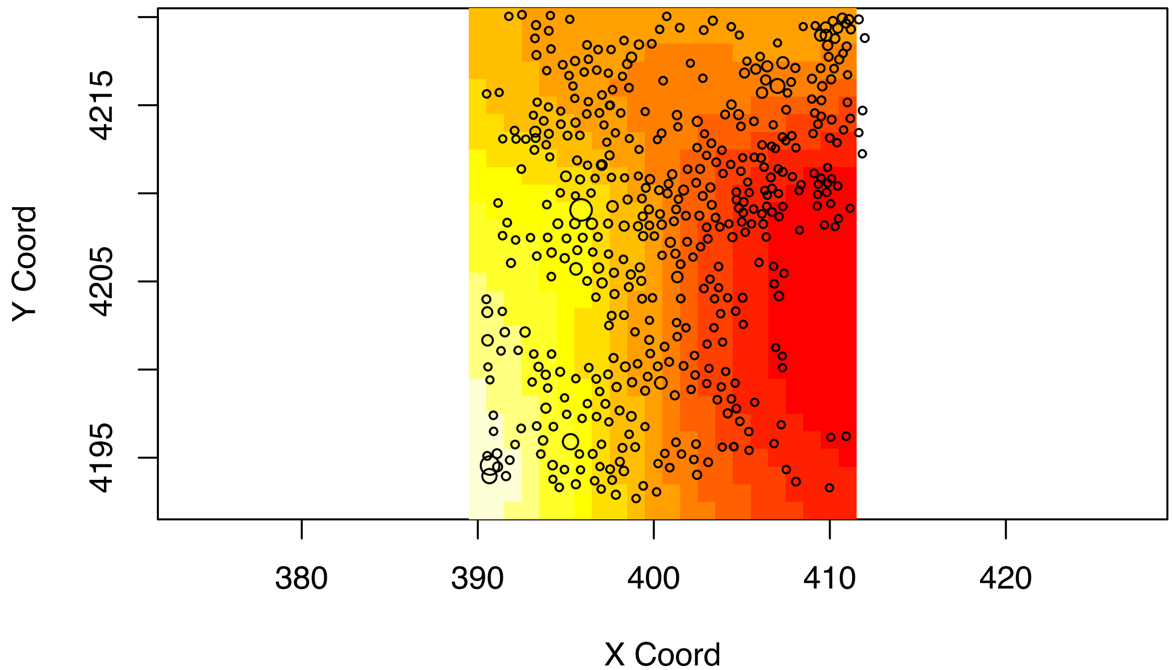
```
y.range<-as.integer(range(b[[1]][,2]))
y.range
```

```
## [1] 4192 4220
```

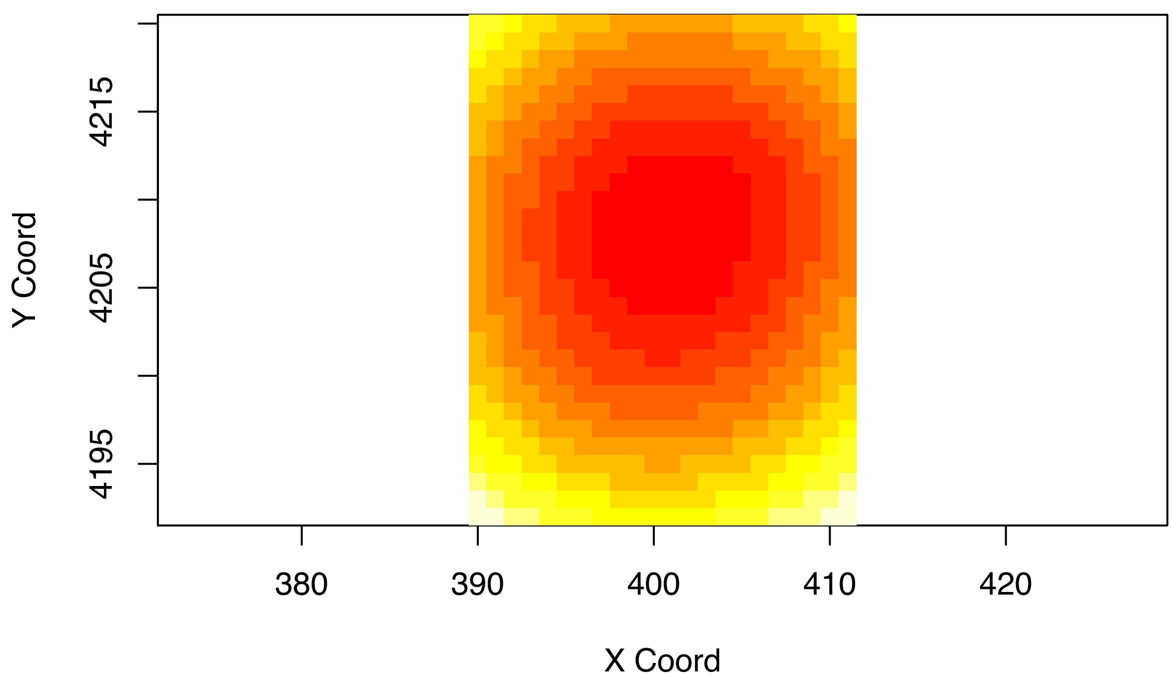
```
grd<-expand.grid(x=seq(from=x.range[1],to=x.range[2],by=1),y=seq(from=y.range[1],to=y.range[2],by=1))
library(sp)
coordinates(grd)<-~x+y

##Different kriggings

#ordinary krigging
q<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd))
```

```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```
#raster map of predict values
image(q,val=q$predict)
points(a,add=T)
```




```r
#raster map of variances
image(q,val=q$krige.var)
```

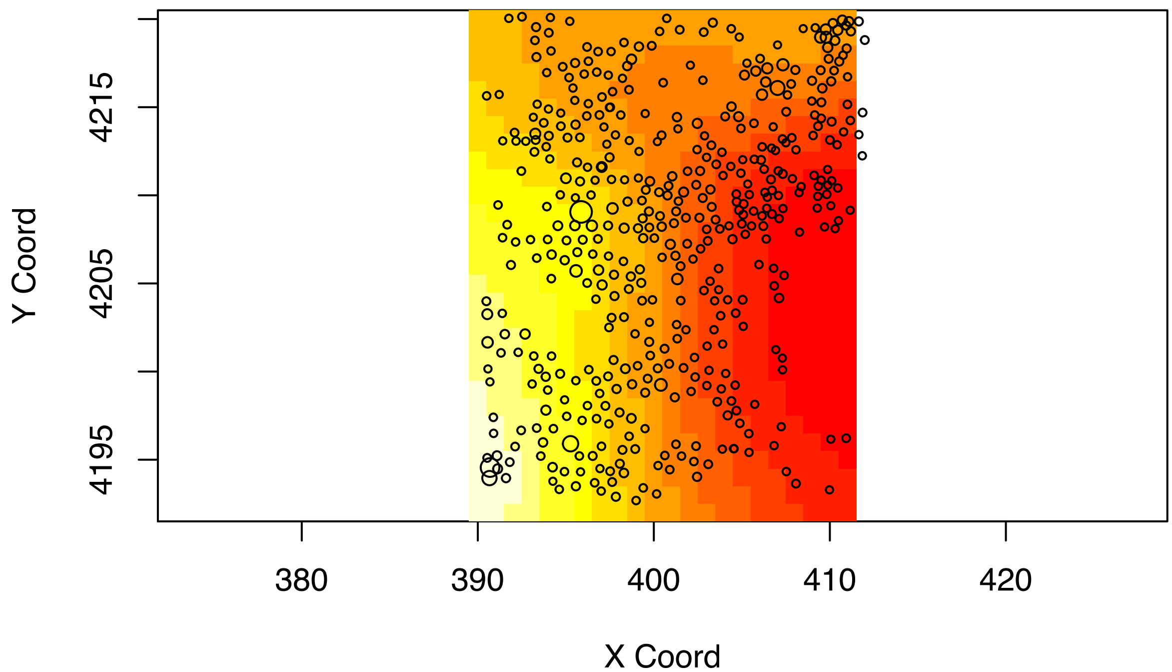



```r
#universal krigging

uq<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd),trend=1)
```

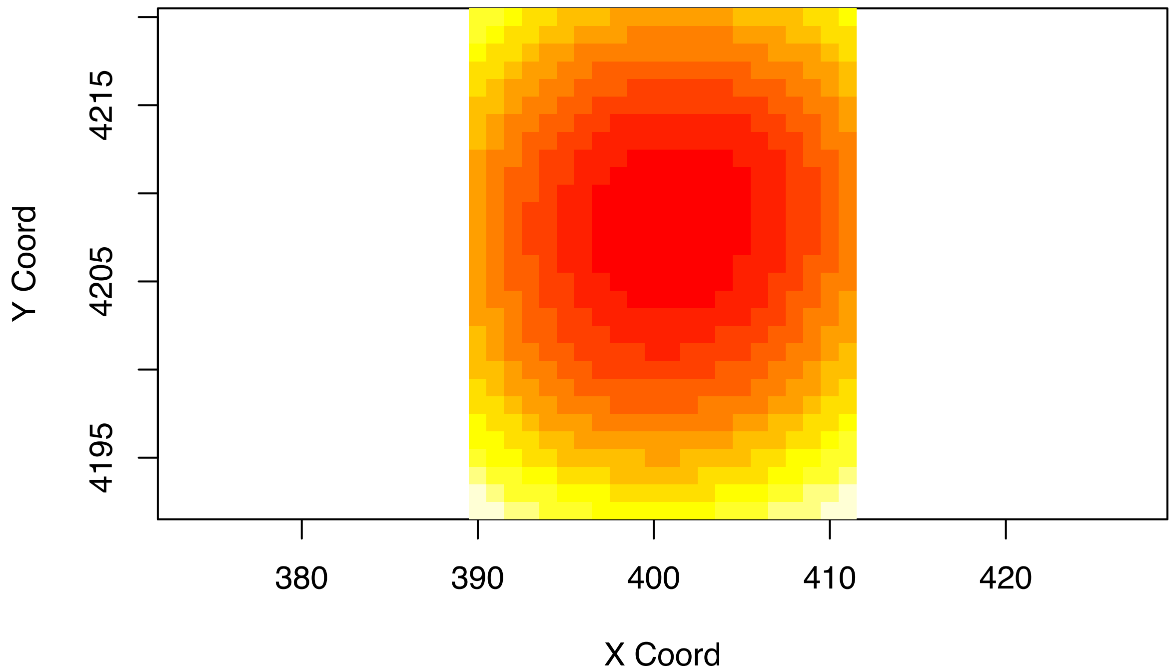
```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
```

```
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```r
#raster map of predict values
image(uq,val=uq$predict)
points(a,add=T)
```




```r
#raster map of variances
image(uq,val=uq$krige.var)
```



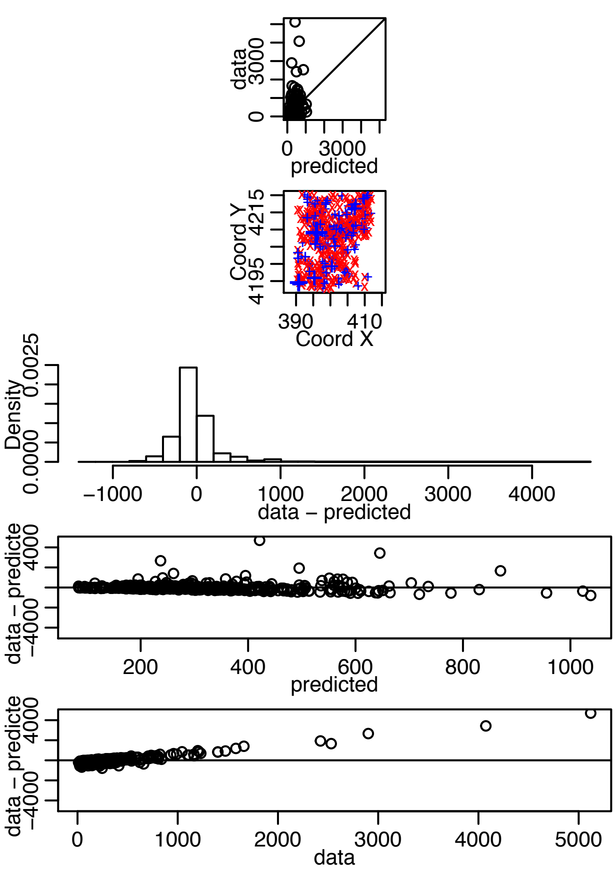

```
#Cross validation using geoR:

#Perform cross validation:
x_val1 <- xvalid(b, model=fit1)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## xvalid: end of cross-validation
```

```
press1 <- sum(x_val1$error^2)

press1
```

```
## [1] 72887569
```

```
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
plot(x_val1)
```




```
#Plot the sample variogram:
plot(var1)

#use sph model
fit2<-variofit(var1,cov.model="sph",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is spherical
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

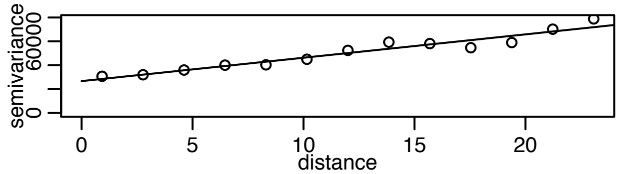
```
#Plot the model variogram:
lines(fit2)

#Perform cross validation:
x_val2 <- xvalid(b, model=fit2)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## xvalid: end of cross-validation
```

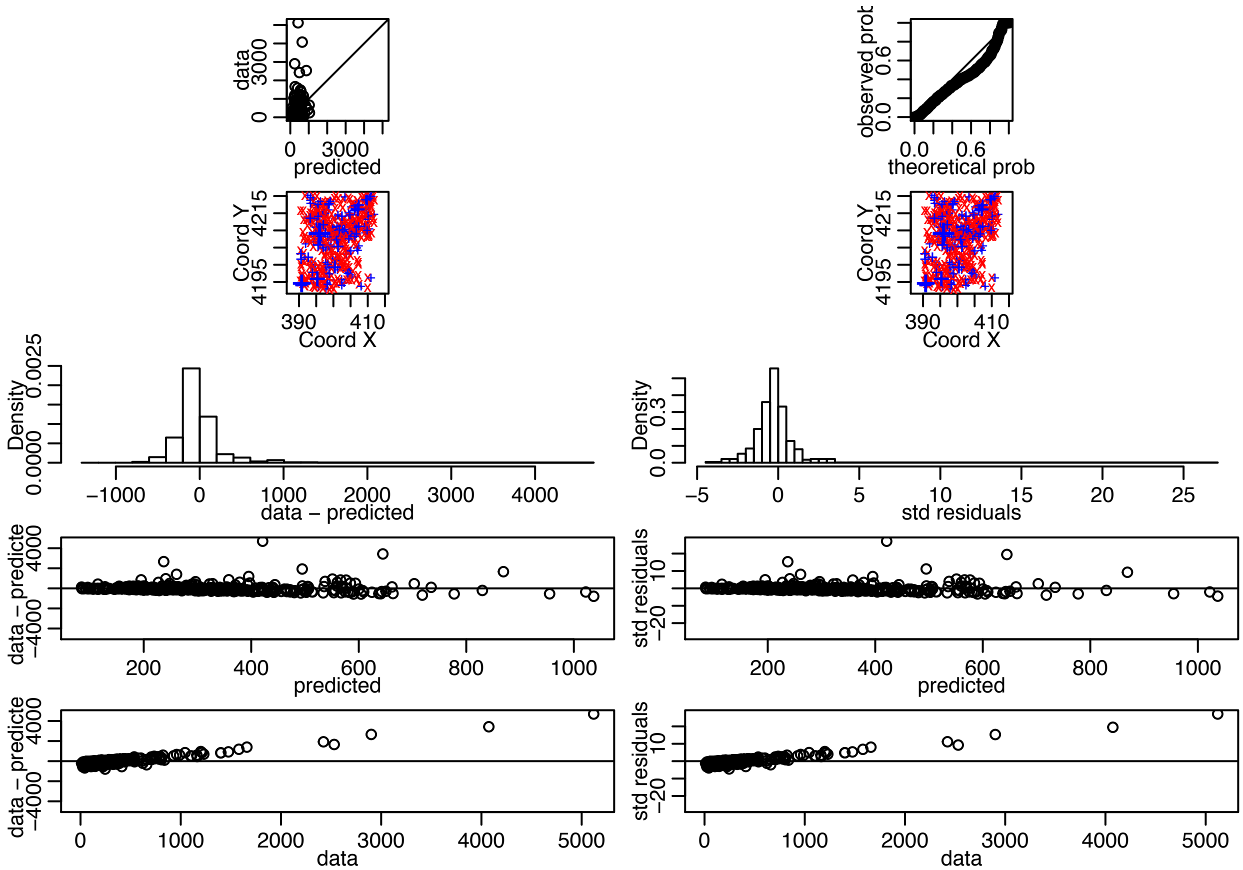
```
press2 <- sum(x_val2$error^2)

press2
```

```
## [1] 72892477
```

```
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
```




```r
plot(x_val2)
```




```r
#======================================================
#Re-estimating the variogram each time a data point is omitted (exp model):
x_val_reest1 <- xvalid(b, model=fit1, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## xvalid: end of cross-validation
```

```r
press1_reest <- sum(x_val_reest1$error^2)

press1_reest
```

```
## [1] 73471359
```

```r
#Re-estimating the variogram each time a data point is omitted (sph model):
x_val_reest2 <- xvalid(b, model=fit2, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
```

```
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

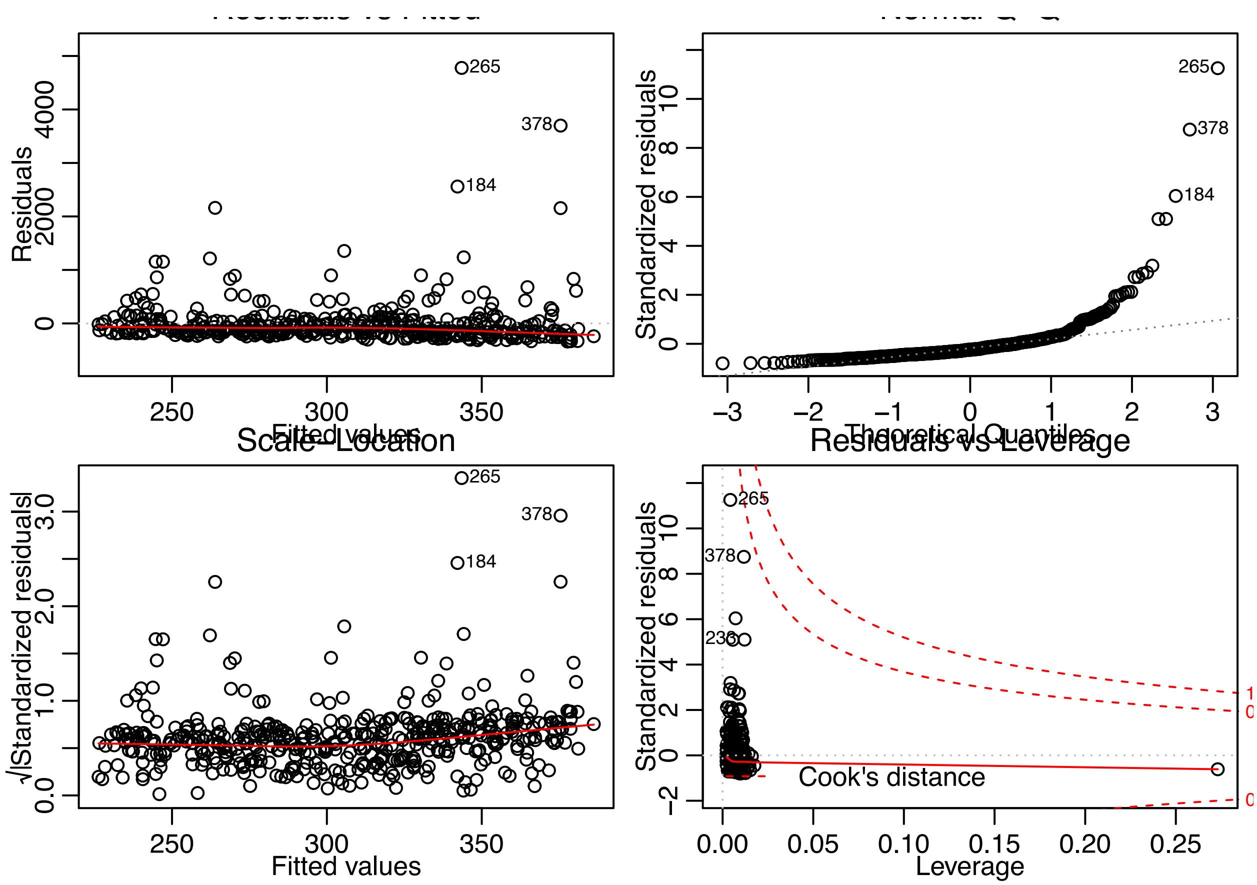
```r
press2_reest <- sum(x_val_reest2$error^2)

press2_reest
```

```
## [1] 73474674
```

```r
#Comparison of the two methods (with and without reestimation of the variogram after omitting a data po
compare <- cbind(x_val1$predicted, x_val2$predicted)
compare[1:5,]
```

```
##          [,1]     [,2]
## [1,] 343.8065 343.7771
## [2,] 178.8938 178.9137
## [3,] 144.9531 145.0516
## [4,] 241.9583 241.9282
## [5,] 372.2451 372.4498
```

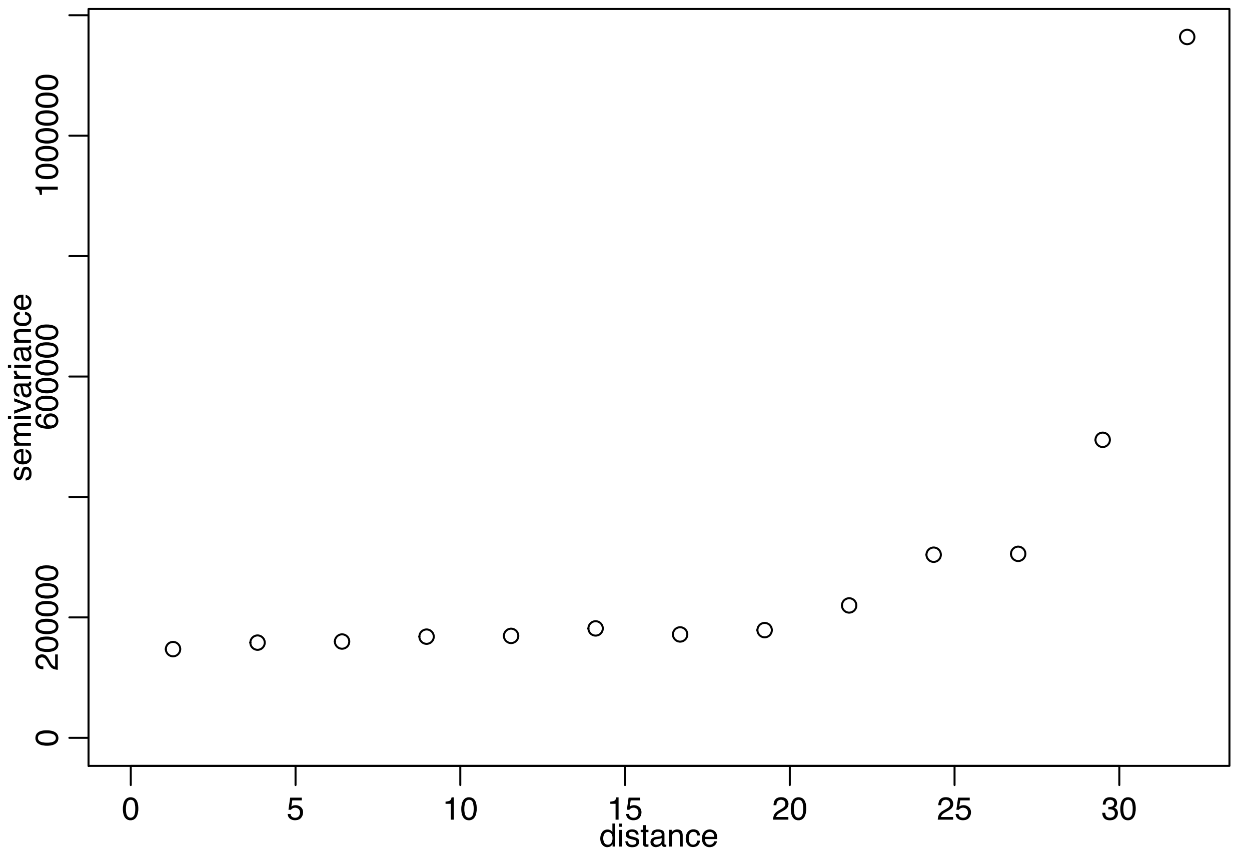
```r
#linear regression analysis
q<-lm(project$Initial_Potential ~ project$easting.Km.+project$northing.Km.)
par(mfrow=c(2,2))
plot(q)
```



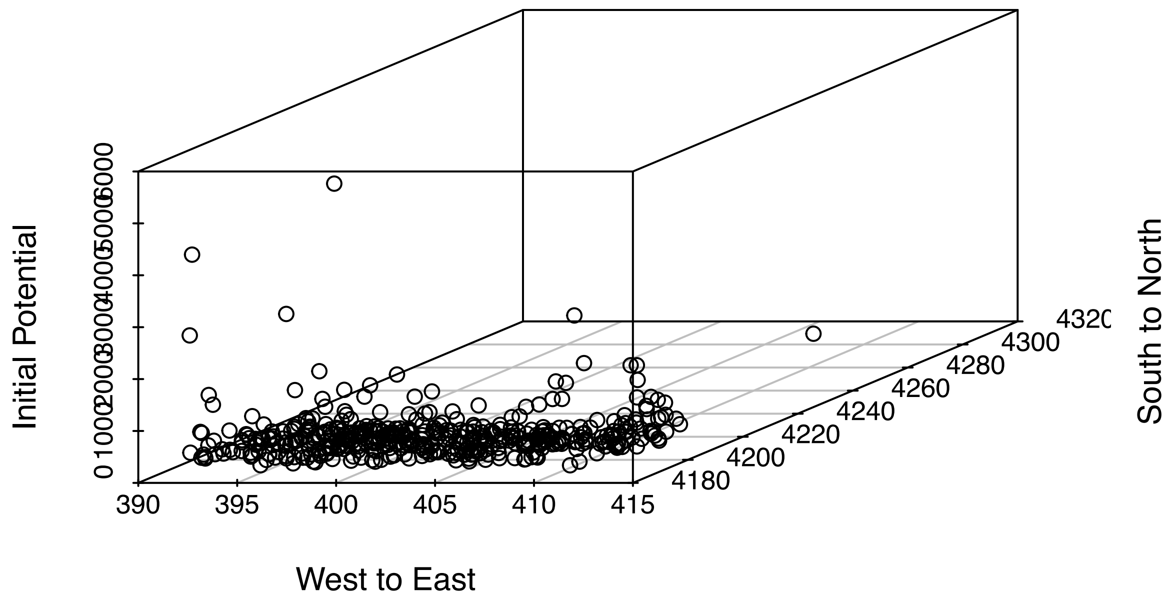

```r
#try to detrend
vario_detrend<-variog(b,trend="1st")
```

```
## variog: computing omnidirectional variogram
```

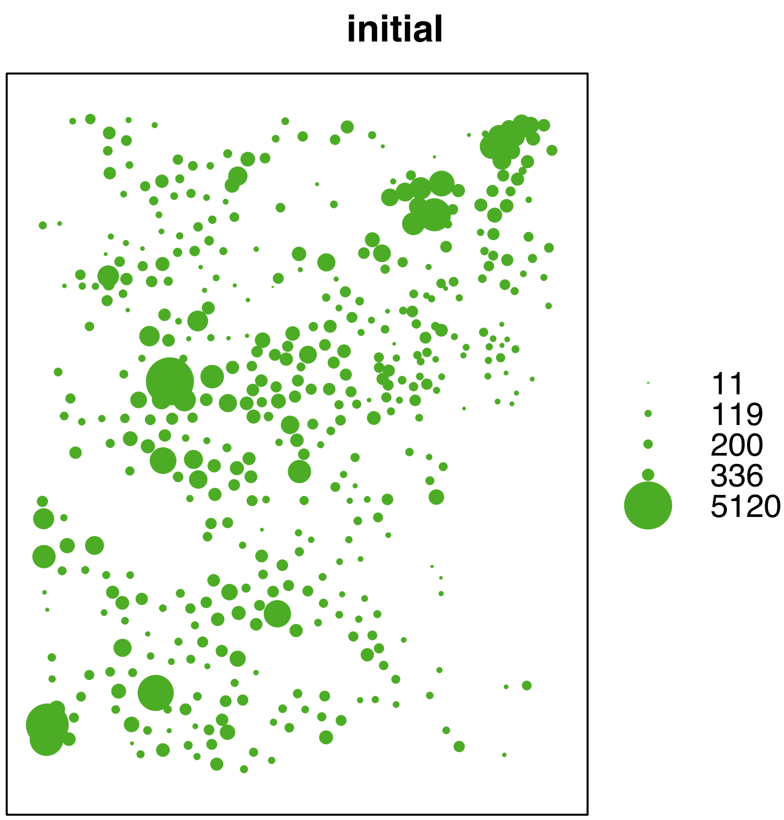
```
par(mfrow=c(1,1))
plot(vario_detrend)
```




```
#plot the 3d data points
library(scatterplot3d)
scatterplot3d(project$easting.Km.,project$northing.Km.,project$Initial_Potential,xlab="West to East",yla
```



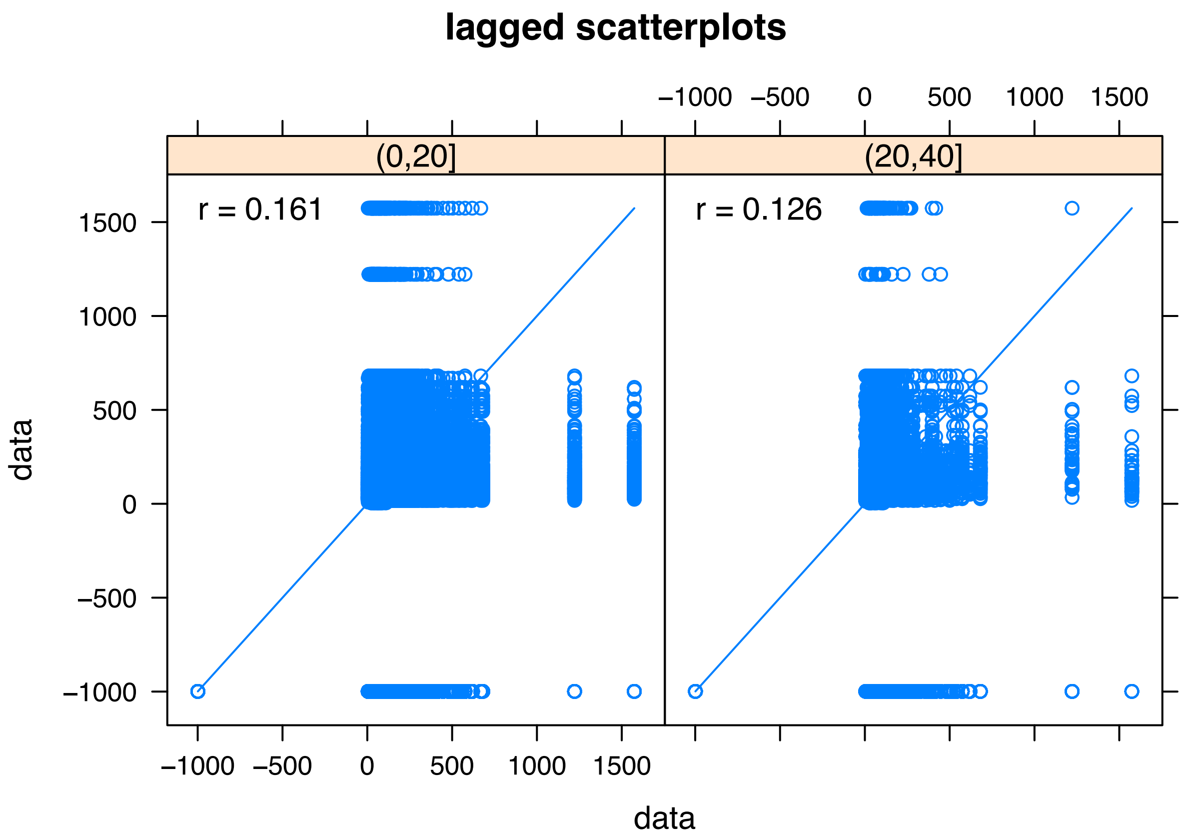
```r
#bubble plots
library(gstat)
library(sp)
b<-data.frame(b)
colnames(b)<-c("x","y","initial")
coordinates(b)<- ~x+y
bubble(b,"initial")
```



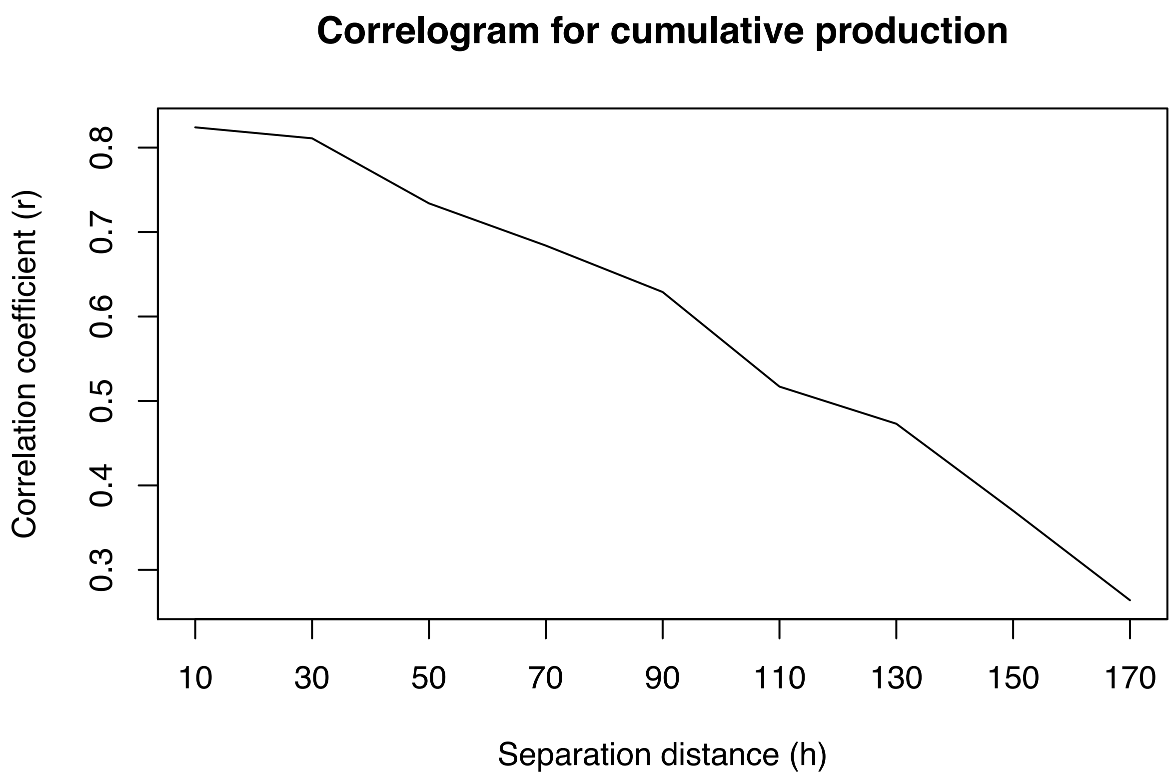
We can see press2 is smaller than press1 so it is better to use spherical model.

Now do the analysis on the cumulative production

```r
extreme=project$northing.Km.>4300
b<-cbind(project$easting.Km.[!extreme],project$northing.Km.[!extreme],project$Cumulative_Production[!ex

##create h-scatterplots
c<-as.data.frame(b)
names(c) <- c("x", "y", "data")
library(gstat)
library(sp)
coordinates(c) <- ~x+y

qq <- hscat(data~1, c, c(0,20,40,60,80,100,120,140,160,180))

plot(qq, main="h-scatterplots")
```



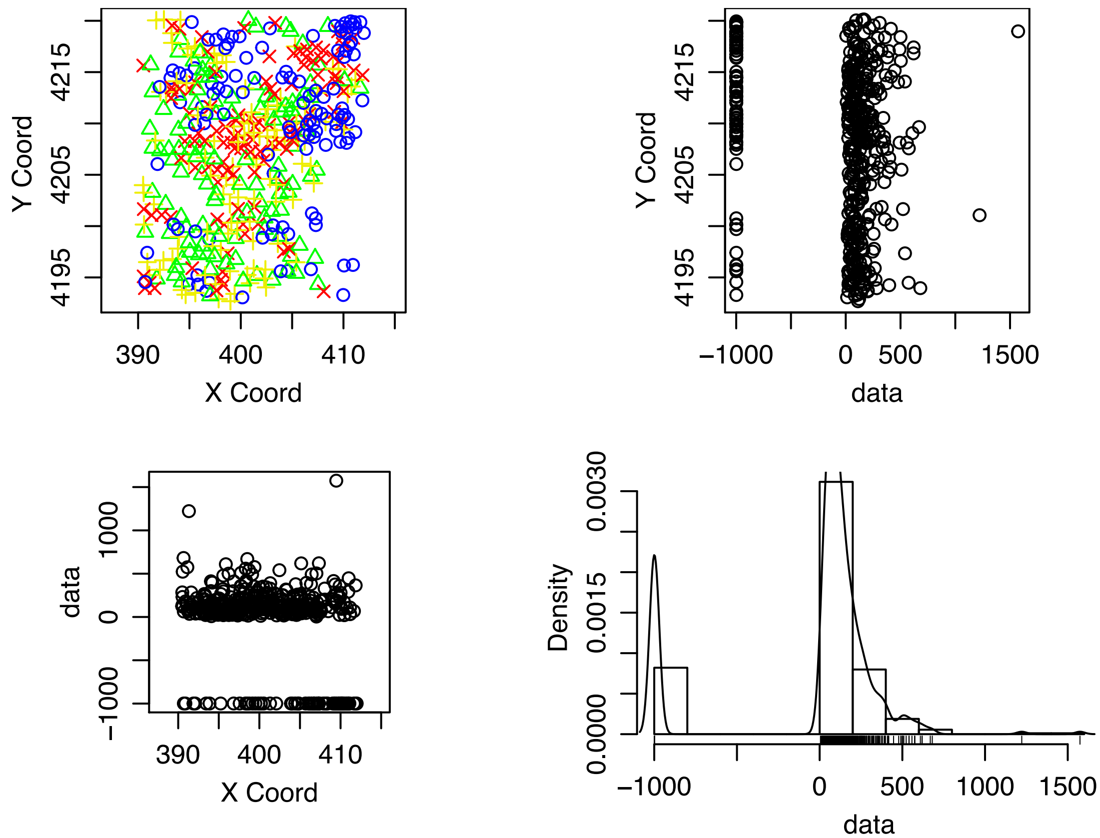

```
#Correlogram for cumulative production
plot(c(10,30,50,70,90,110,130,150,170), c(0.824, 0.811,0.734,0.684,0.629,0.517,0.473,0.37, 0.264), type=

axis(1, at=seq(10, 190, by=20),labels=seq(10, 190, by=20))

axis(2, at=seq(0, 1, by=0.1),labels=seq(0, 1, by=0.1))
```




```
b<-as.geodata(b)
#The graph without outliers
summary(b)
```

```
## Number of data points: 451
##
## Coordinates summary
##      Coord1   Coord2
## min 390.489 4192.684
## max 411.986 4220.133
##
## Distance summary
##         min         max
##  0.02109502 33.34321307
##
## Data summary
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -999.00000   35.00000   96.00000  -28.99335  183.50000 1574.00000
```

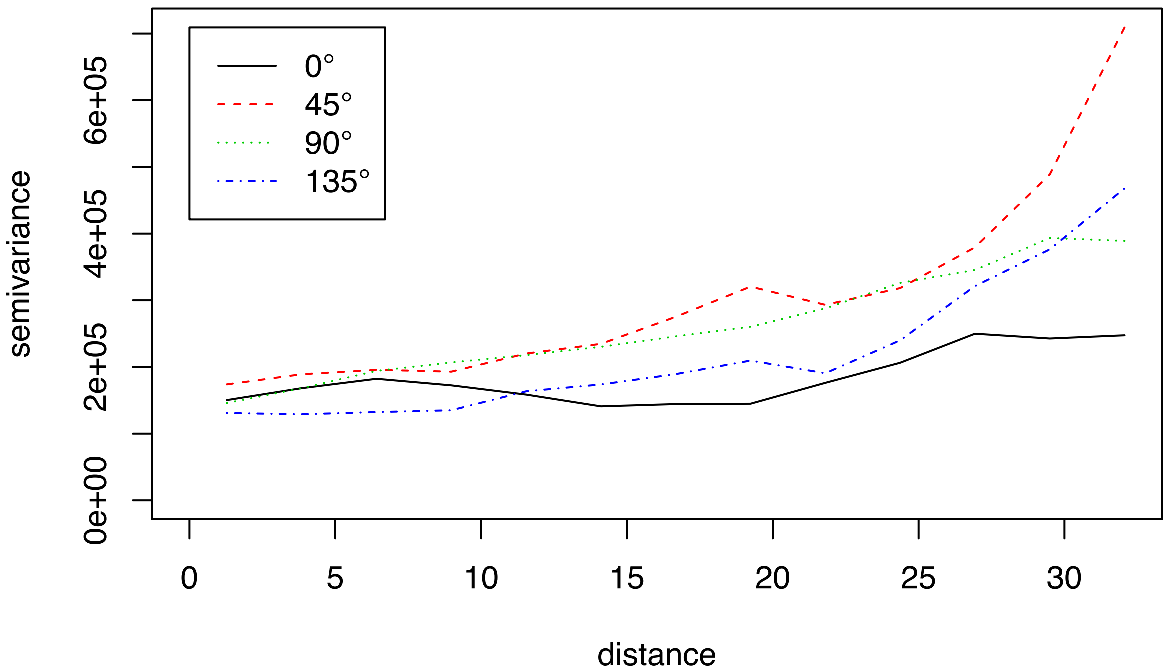
```
plot(b)
```




```r
#plot the cariograms in for different directions
var1<-variog4(b)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 90 degrees (1.571 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 135 degrees (2.356 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing omnidirectional variogram
```

```r
plot(var1)
```




```
#Compute the sample variogram and fit a model to it:
var1<-variog(b,dir=pi/4,estimator.type="modulus",max.dist = 24)
```

```
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
```

```
#Plot the sample variogram:
plot(var1)

fit1<-variofit(var1,cov.model="exp",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
#Obtain the estimates of the parameters:
fit1
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
## parameter estimates:
##        tausq      sigmasq          phi
##   19488.6240 3859300.2266     477.4251
## Practical Range with cor=0.05 for asymptotic range: 1430.238
##
## variofit: minimised weighted sum of squares = 8.791214e+12
```

```
#Plot the model variogram:
lines(fit1)
initial.values<-expand.grid(seq(10,100,by=10),seq(10000,15000,by=200))
#npairs weights
fit1l<-variofit(var1,cov.model="exp",ini.cov.pars=initial.values,wei="npairs", fix.nugget=FALSE, nugget=
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
```

```
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 363818277903168
```

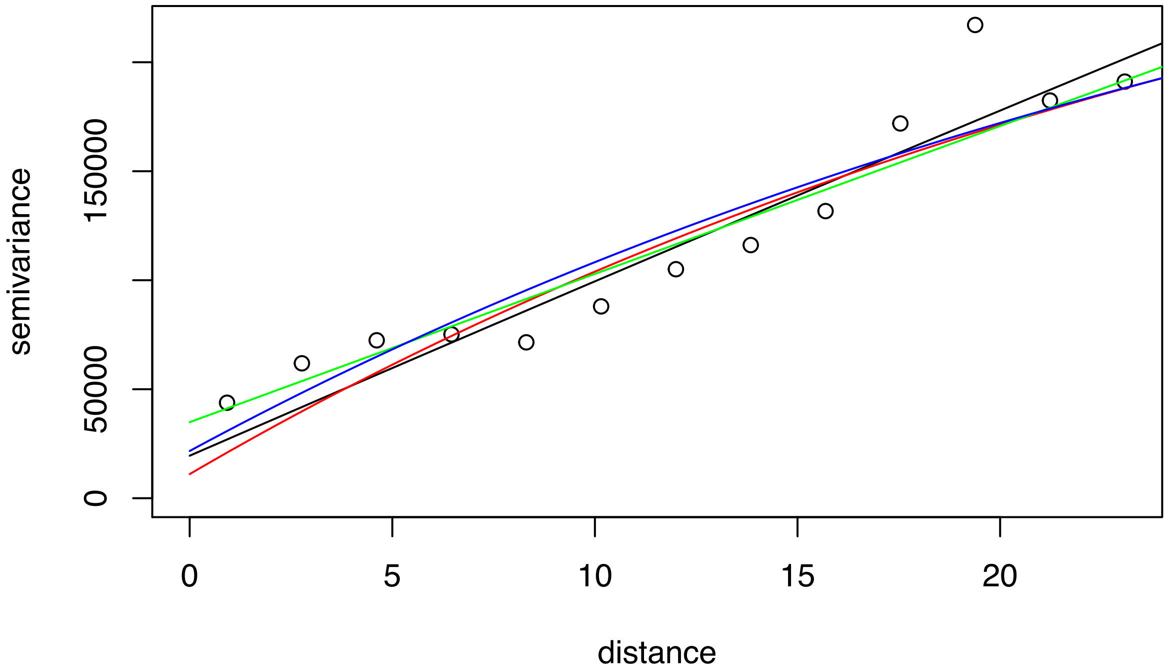
```
lines(fit1l,lty=1,col="red")
#cressies weights
fit2l<-variofit(var1,cov.model="exp",weights="cressie",ini.cov.pars=initial.values,fix.nugget=FALSE,nugg
```

```
## variofit: covariance model used is exponential
## variofit: weights used: cressie
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 909530.070640911
```

```
lines(fit2l,lty=1,col="green")
#equal weights (simply OLS):
fit3l<-variofit(var1,cov.model="exp", ini.cov.pars=initial.values,weights="equal",fix.nugget=FALSE,nugge
```

```
## variofit: covariance model used is exponential
## variofit: weights used: equal
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "100"   "10000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 161940801327.291
```

```
lines(fit3l,lty=1,col="blue")
```



```r
#create grid
x.range<-as.integer(range(b[[1]][,1]))
x.range
```

```
## [1] 390 411
```

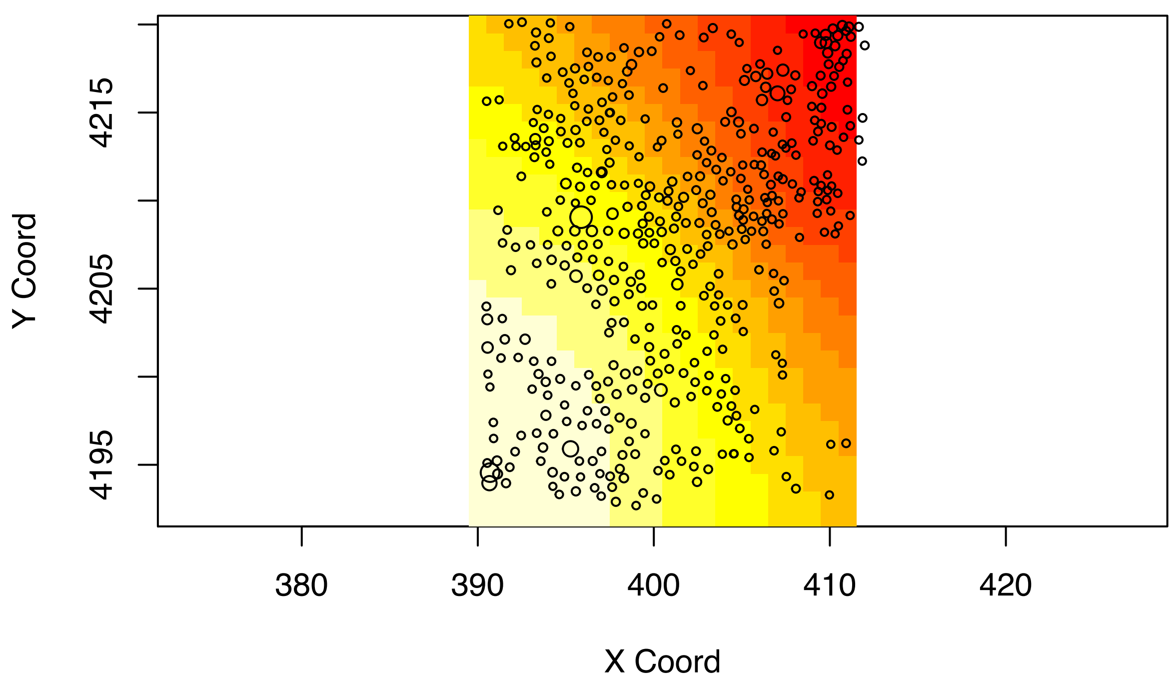
```r
y.range<-as.integer(range(b[[1]][,2]))
y.range
```

```
## [1] 4192 4220
```

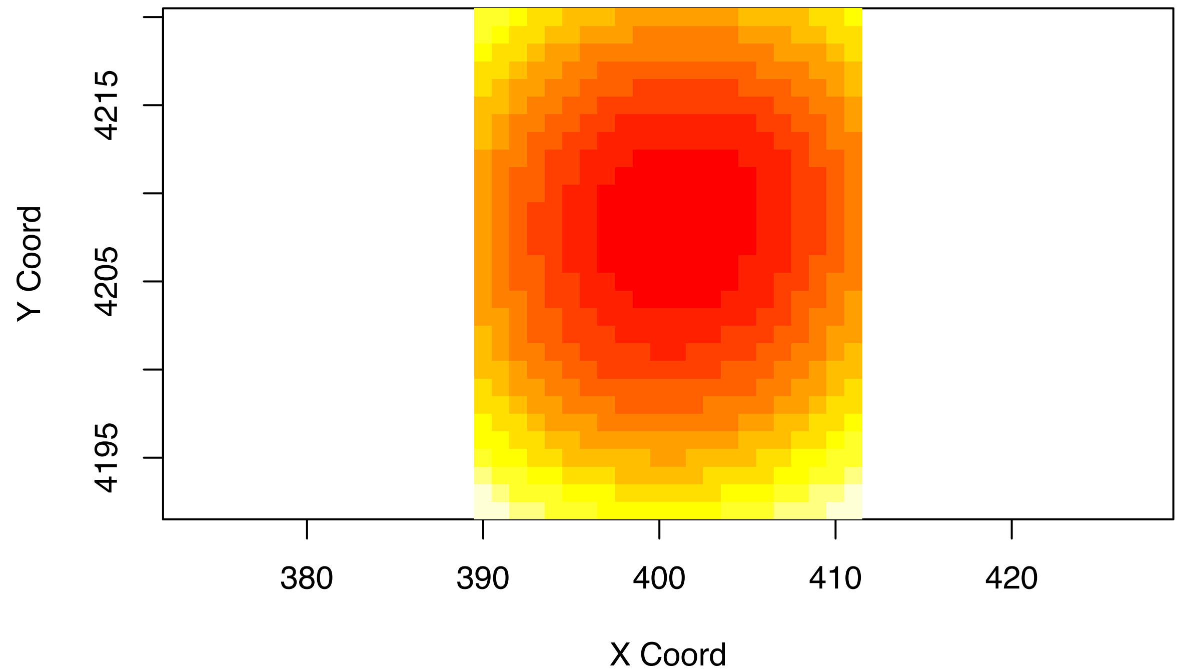
```r
grd<-expand.grid(x=seq(from=x.range[1],to=x.range[2],by=1),y=seq(from=y.range[1],to=y.range[2],by=1))
library(sp)
coordinates(grd)<-~x+y
#ordinary kriging
q<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd))
```

```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```r
#raster map of predict values
image(q,val=q$predict)
points(a,add=T)
```



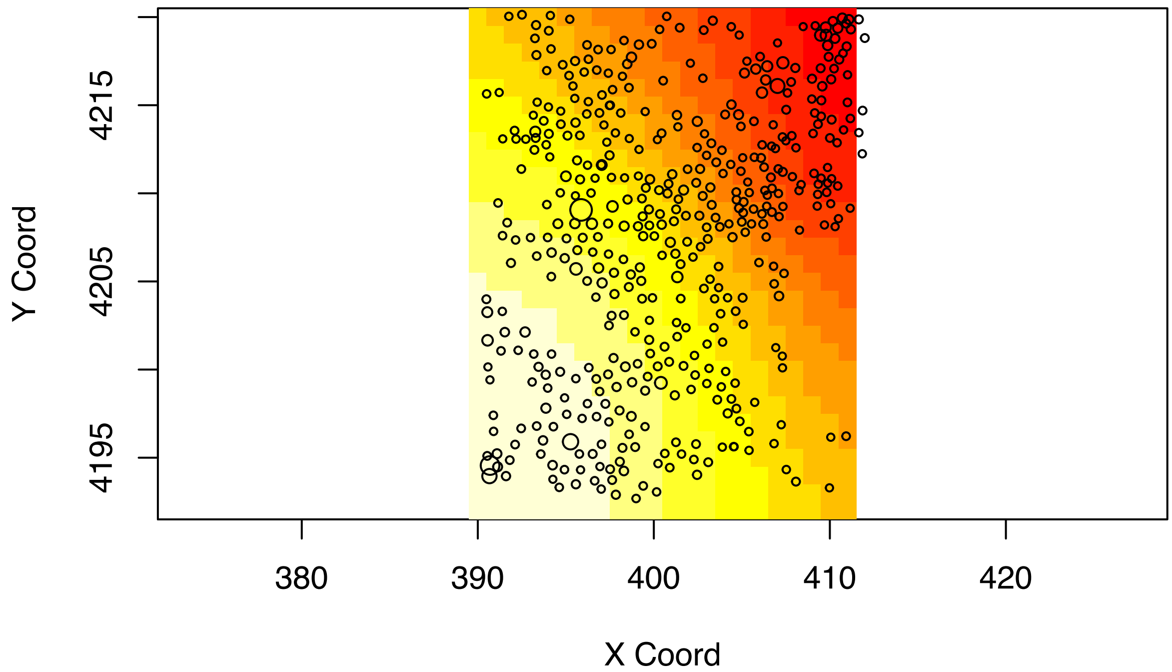

```r
#raster map of variances
image(q,val=q$krige.var)
```



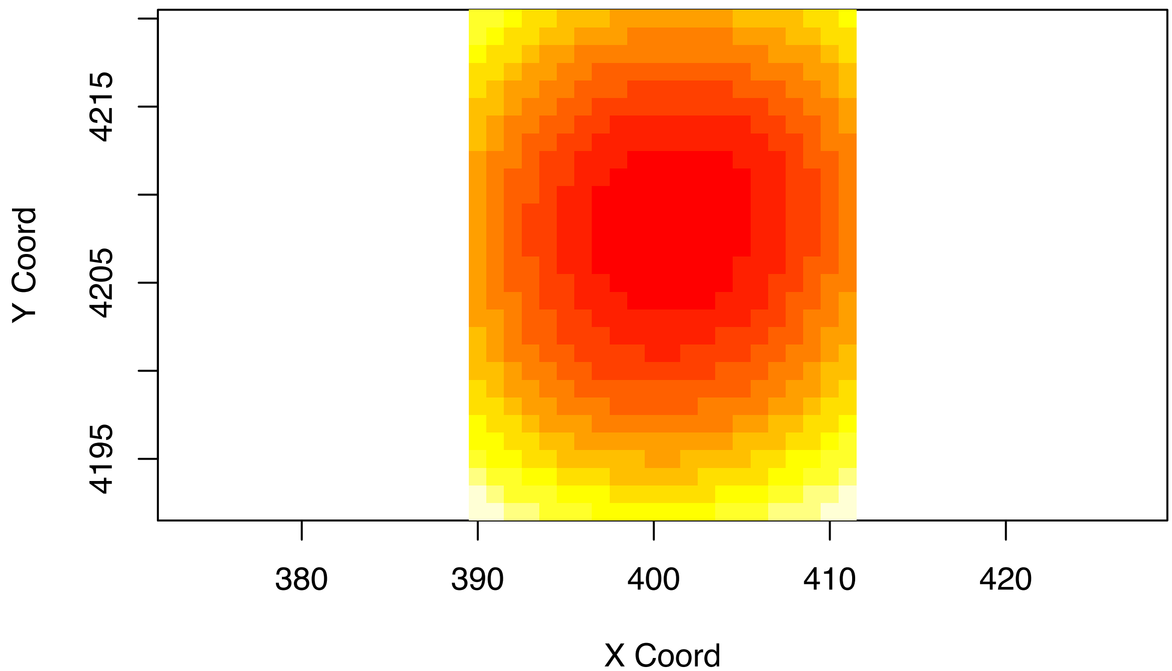

```r
#universal krigging

uq<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd),trend=1)
```

```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```r
#raster map of predict values
image(uq,val=uq$predict)
points(a,add=T)
```




```
#raster map of variances
image(uq,val=uq$krige.var)
```




```
#Cross validation using geoR:

#Perform cross validation:
x_val1 <- xvalid(b, model=fit1)
```

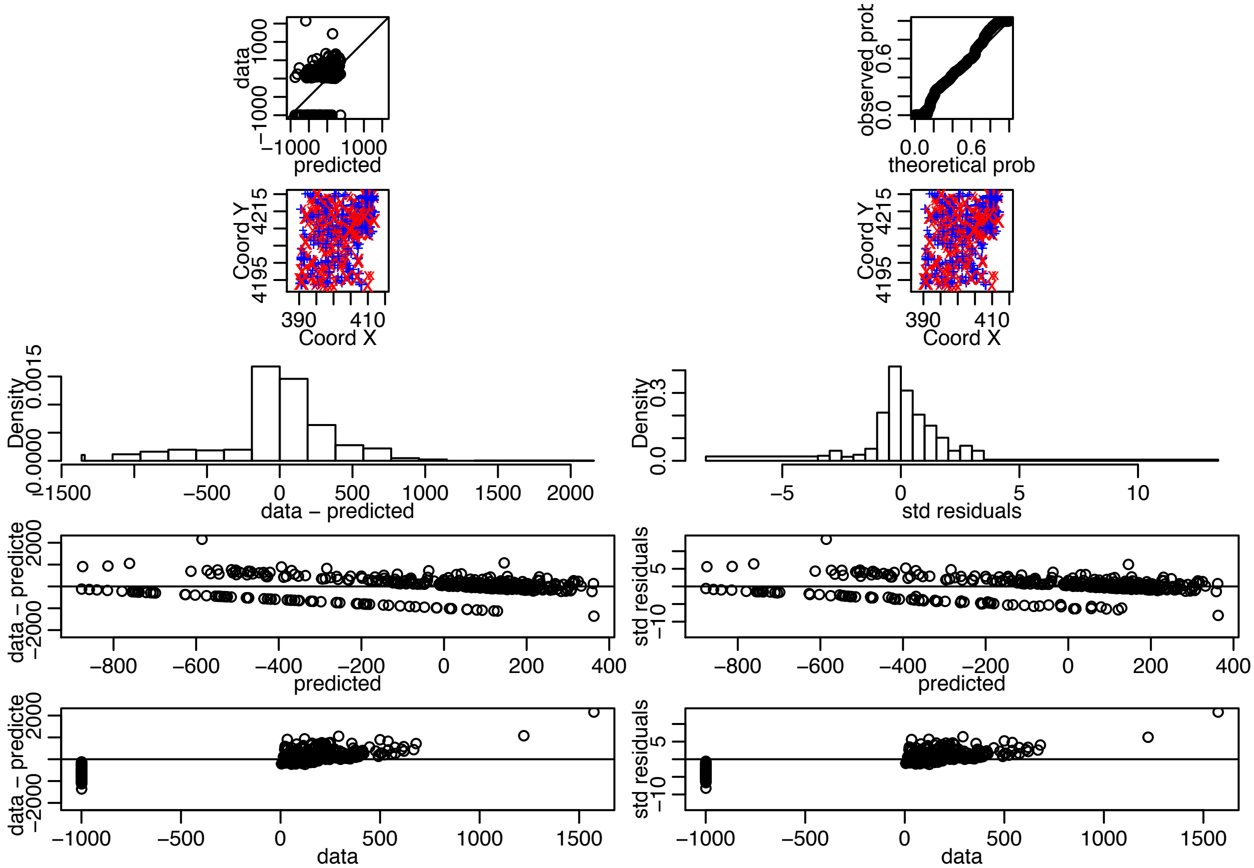
```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
```

```
## xvalid: end of cross-validation
```

```r
press1 <- sum(x_val1$error^2)

press1
```

```
## [1] 66015131
```

```r
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
plot(x_val1)
```




```r
#Plot the sample variogram:
plot(var1)

#use sph model
fit2<-variofit(var1,cov.model="sph",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is spherical
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
#Plot the model variogram:
lines(fit2)

#Perform cross validation:
x_val2 <- xvalid(b, model=fit2)
```

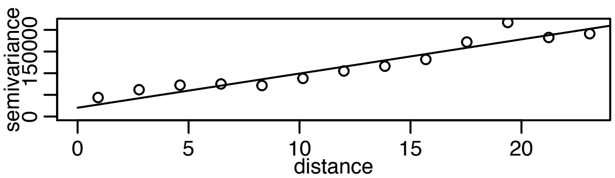
```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
```

```
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

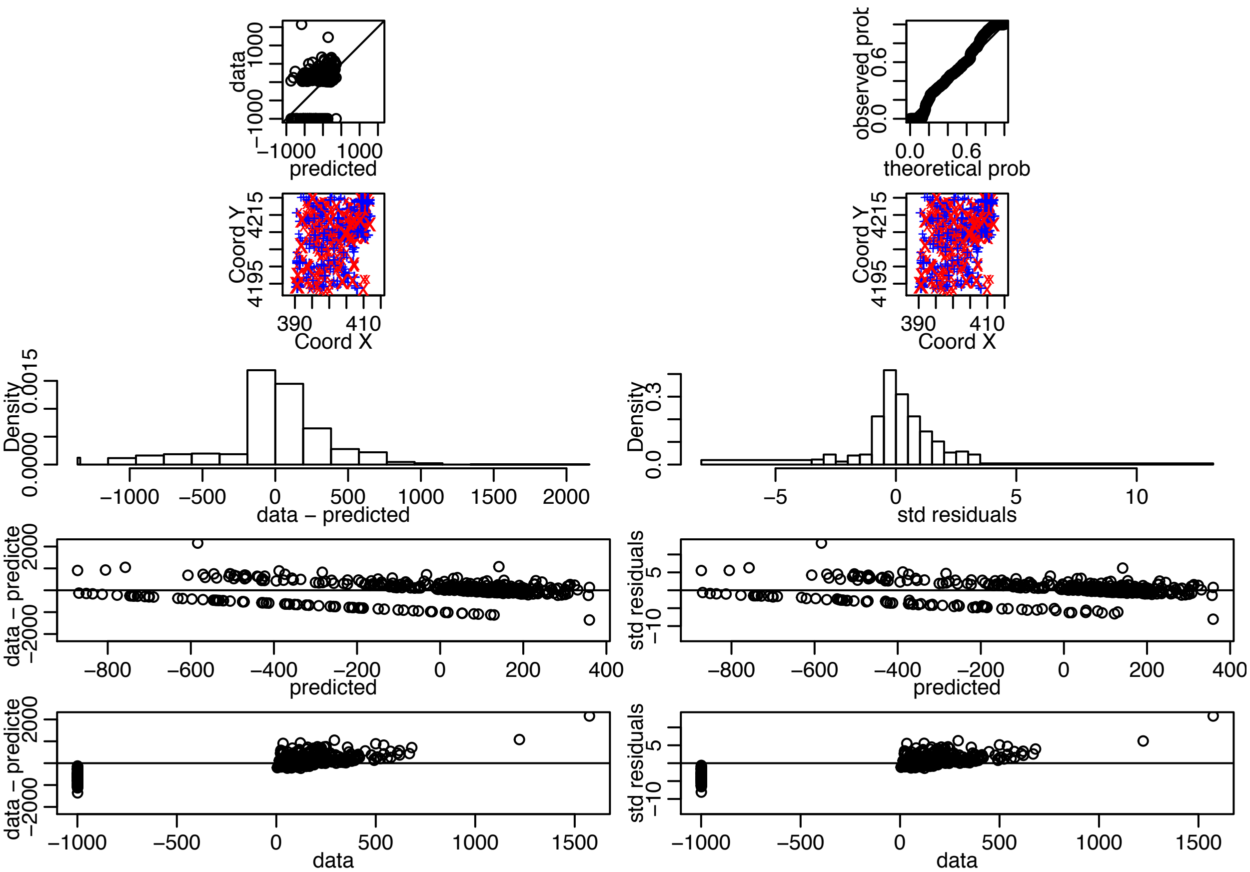
```r
press2 <- sum(x_val2$error^2)

press2
```

```
## [1] 65956292
```

```r
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
```




```r
plot(x_val2)
```




```r
#====================================================
#Re-estimating the variogram each time a data point is omitted (exp model):
x_val_reest1 <- xvalid(b, model=fit1, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

```r
press1_reest <- sum(x_val_reest1$error^2)

press1_reest
```

```
## [1] 66070818
```

```r
#Re-estimating the variogram each time a data point is omitted (sph model):
x_val_reest2 <- xvalid(b, model=fit2, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 451
## xvalid: number of validation locations = 451
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## xvalid: end of cross-validation
```

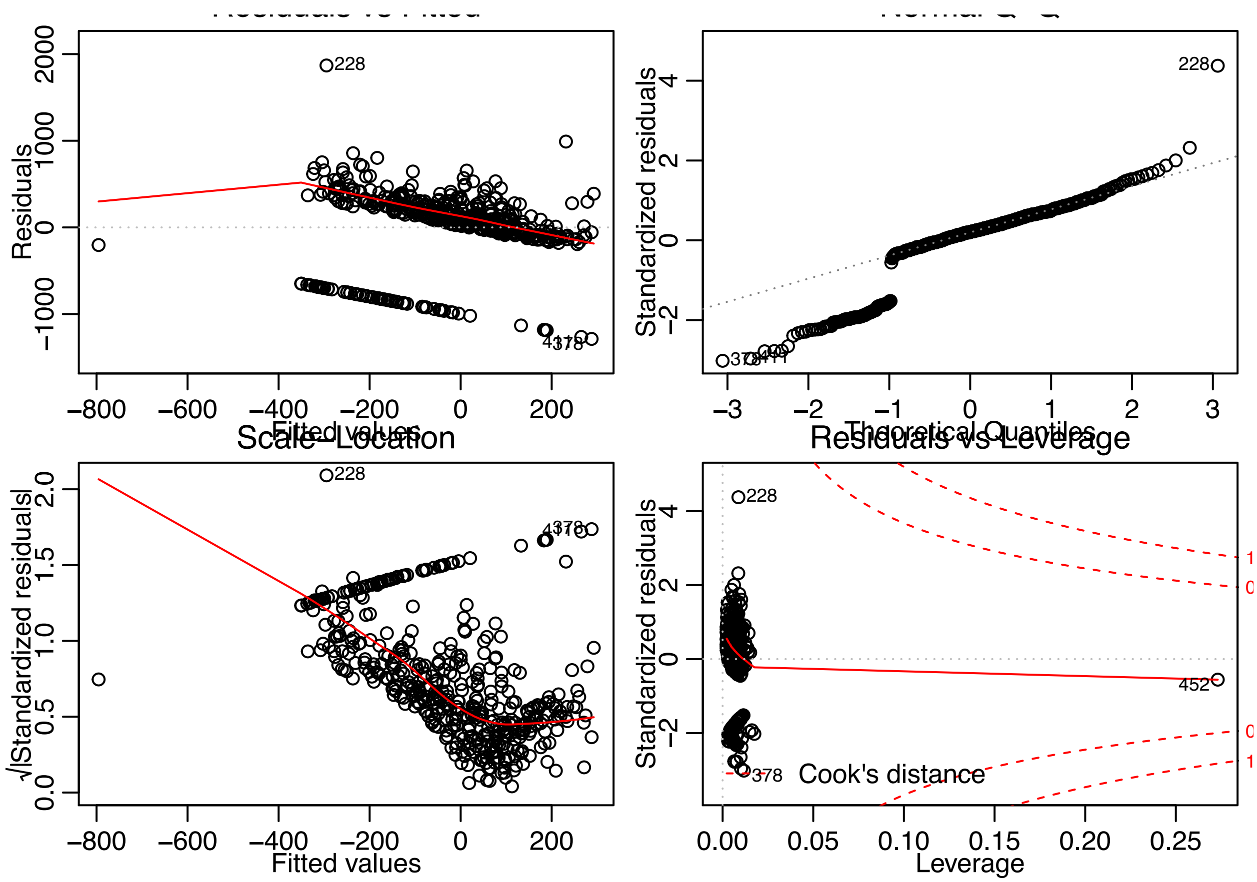
```r
press2_reest <- sum(x_val_reest2$error^2)

press2_reest
```

```
## [1] 65995882
```

```r
#Comparison of the two methods (with and without reestimation of the variogram after omitting a data po
compare <- cbind(x_val1$predicted, x_val2$predicted)
compare[1:5,]
```

```
##            [,1]       [,2]
## [1,] -287.14758 -282.88777
## [2,]   70.90374   70.34761
## [3,]   48.94543   48.62370
## [4,]   85.35776   83.59841
## [5,]   27.54696   27.48320
```

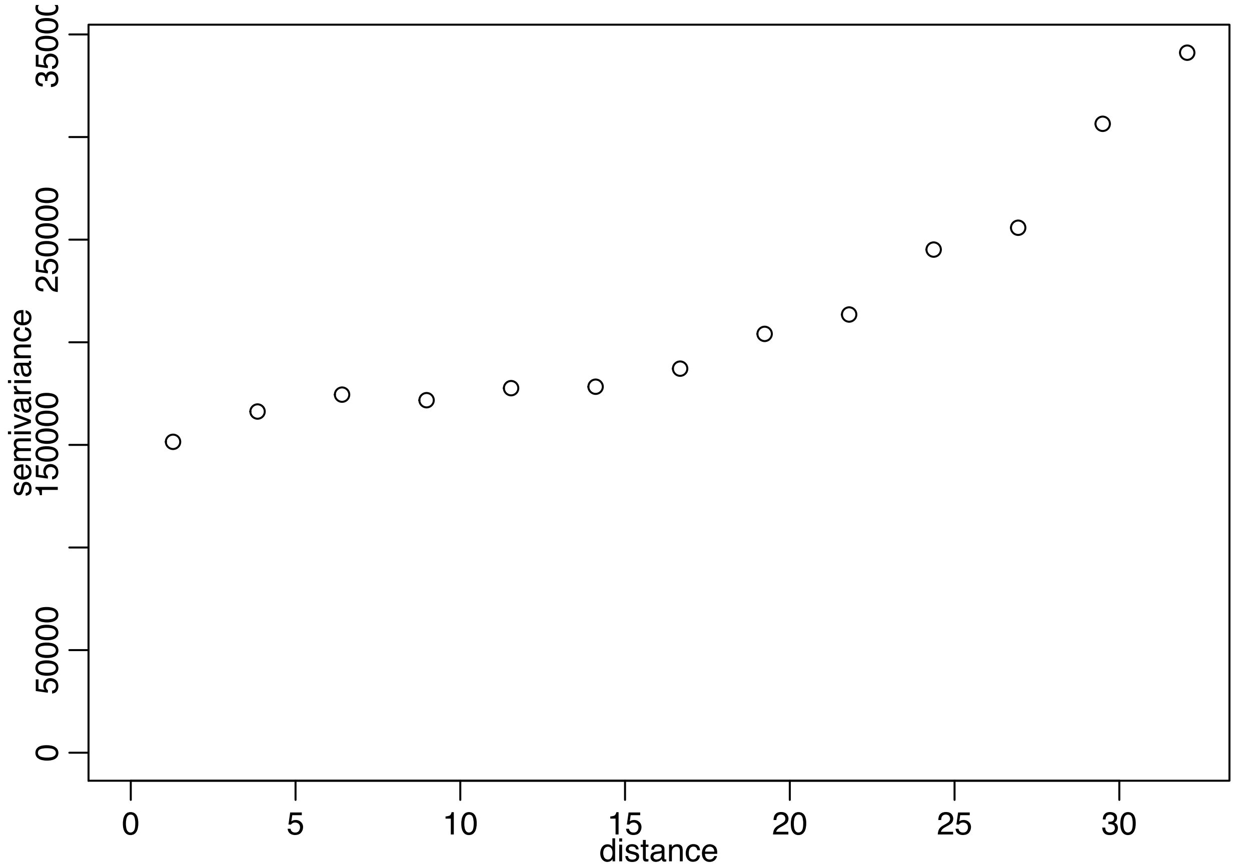
```r
#linear regression analysis
q<-lm(project$Cumulative_Production ~ project$easting.Km.+project$northing.Km.)
par(mfrow=c(2,2))
plot(q)
```



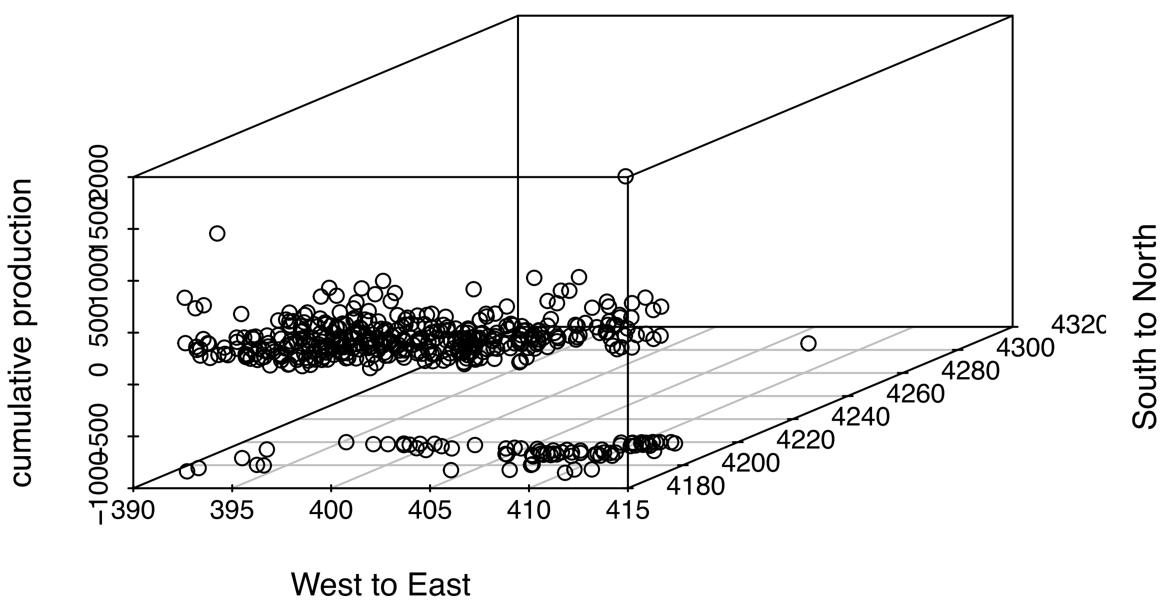

```
#try to detrend
vario_detrend<-variog(b,trend="1st")
```

```
## variog: computing omnidirectional variogram
```

```
par(mfrow=c(1,1))
plot(vario_detrend)
```

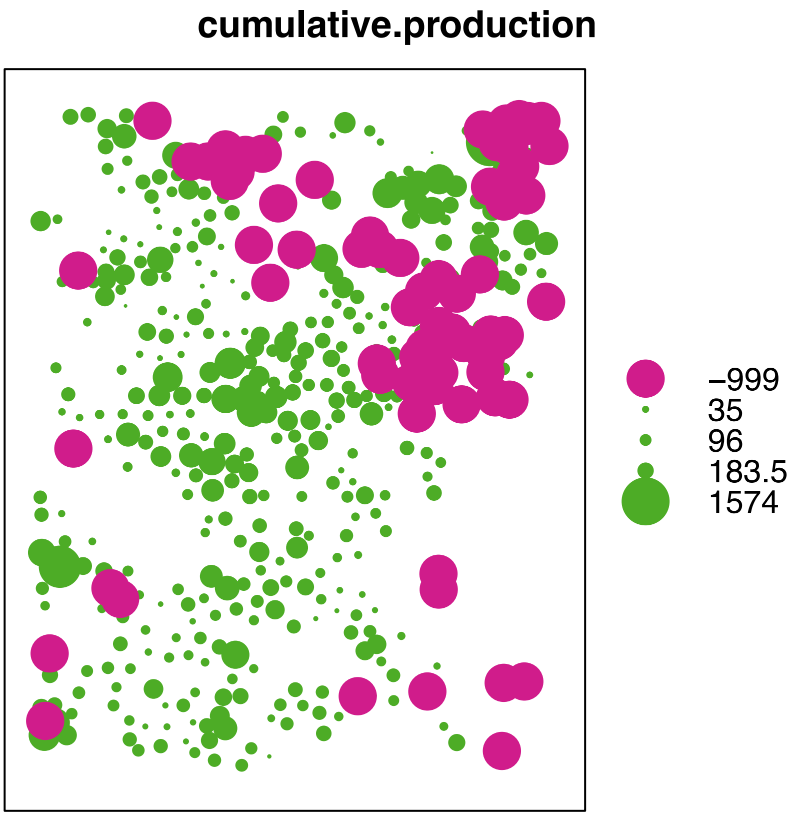



```
#plot the 3d data points
library(scatterplot3d)
scatterplot3d(project$easting.Km.,project$northing.Km.,project$Cumulative_Production,xlab="West to East
```




```
#bubble plots
library(gstat)
library(sp)
```

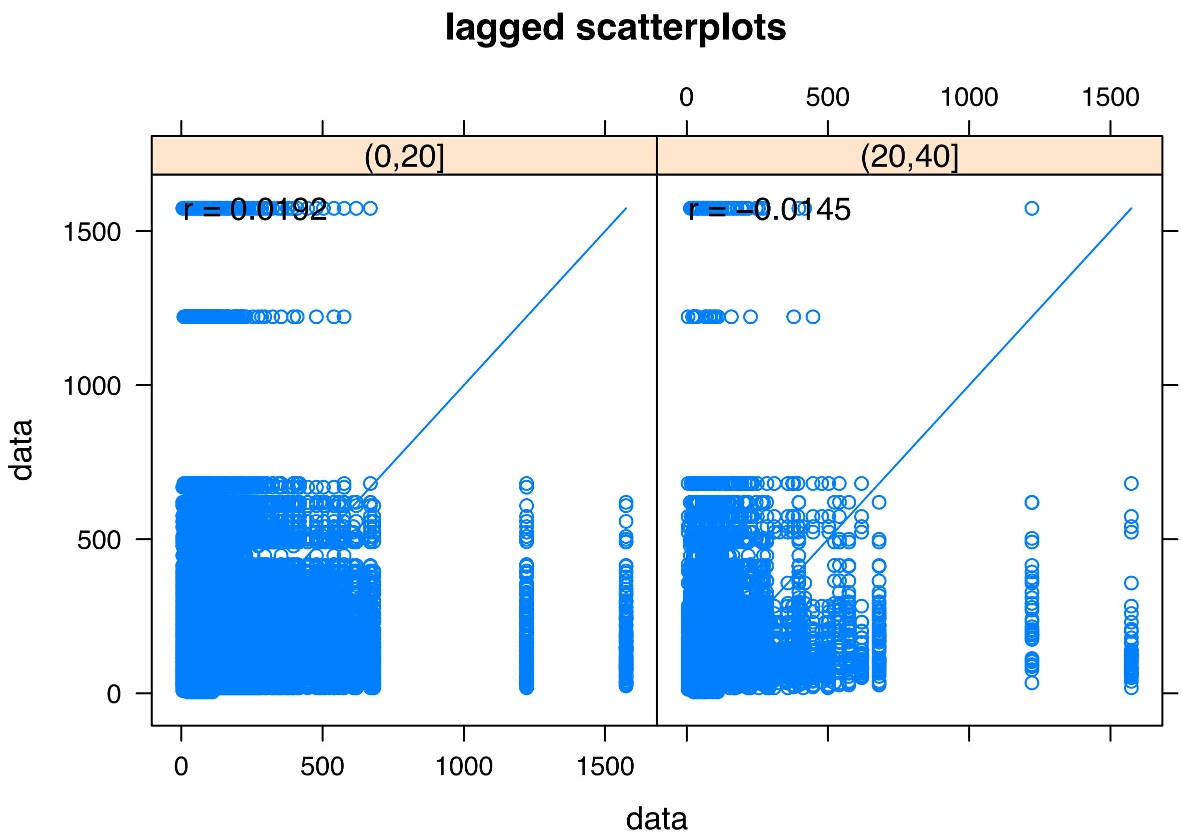
```
b<-data.frame(b)
colnames(b)<-c("x","y","cumulative.production")
coordinates(b)<- ~x+y
bubble(b,"cumulative.production")
```



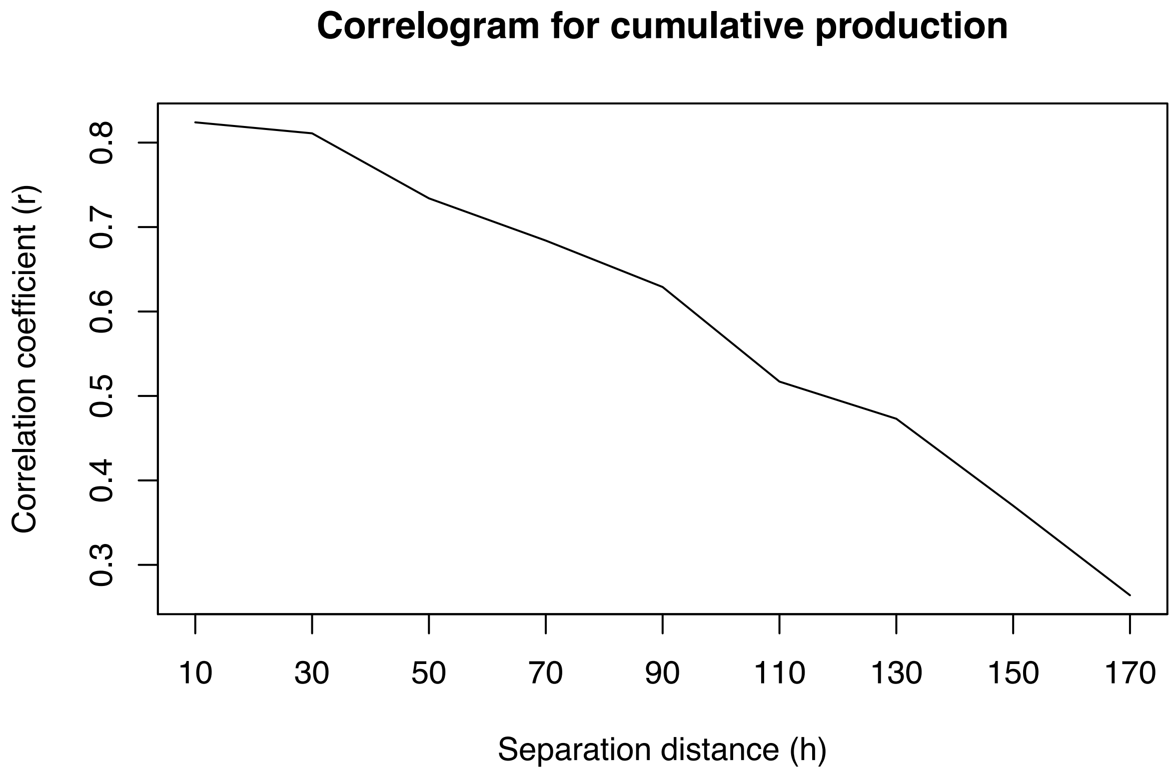
We can see press2 is smaller than press1 so it is better to use spherical model.

Look through the dataset, I notice there are cumulative preoduction data points that are -999, which does not make sense. In the website I found the dataset,it does not have clear explanation on the negative values. I think it is very likely that the negateive values represent that it stops petroleum production at that location. So I excludes those data points and make another analysis about the stations that continue produce petroleum.

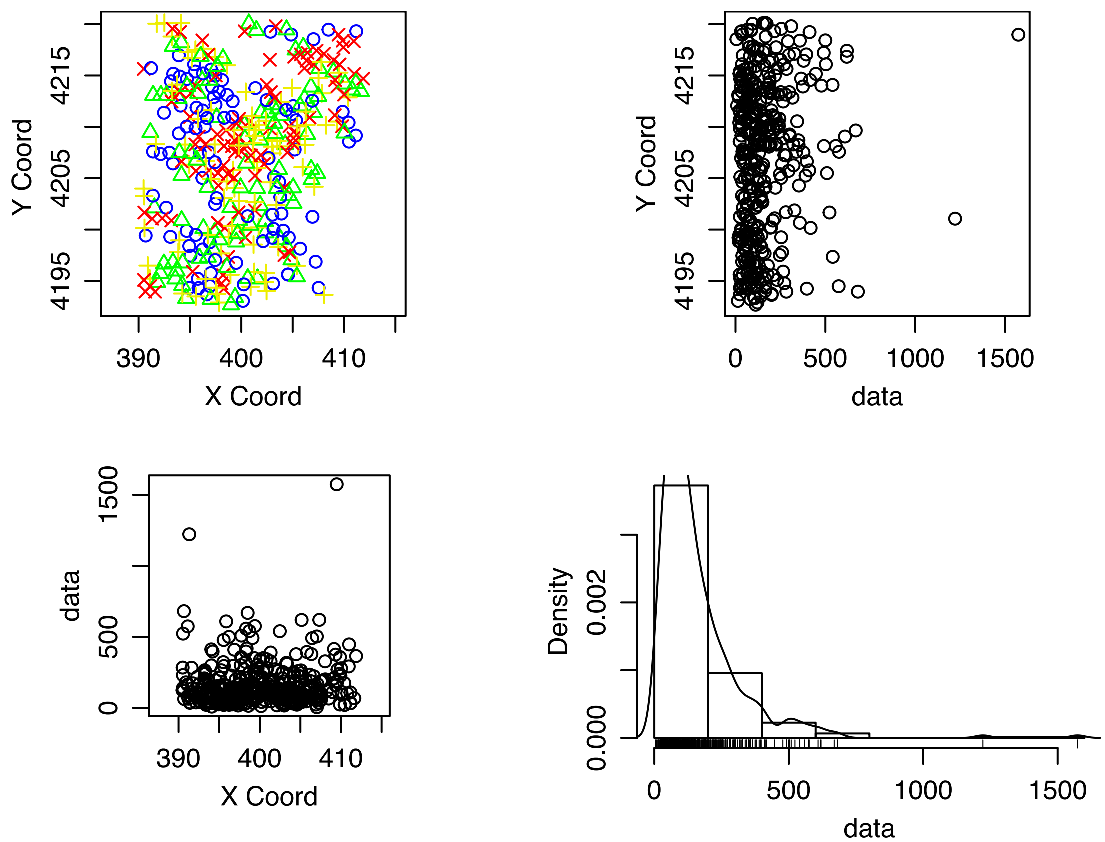
```
update<-project$Cumulative_Production[project$Cumulative_Production>0]
x<-project$easting.Km.[project$Cumulative_Production>0]
y<-project$northing.Km.[project$Cumulative_Production>0]
b<-cbind(x,y,update)
##create h-scatterplots
c<-as.data.frame(b)
names(c) <- c("x", "y", "data")
library(gstat)
library(sp)
coordinates(c) <- ~x+y

qq <- hscat(data~1, c, c(0,20,40,60,80,100,120,140,160,180))

plot(qq, main="h-scatterplots")
```




```
#Correlogram for cumulative production
plot(c(10,30,50,70,90,110,130,150,170), c(0.824, 0.811,0.734,0.684,0.629,0.517,0.473,0.37, 0.264), type=

axis(1, at=seq(10, 190, by=20),labels=seq(10, 190, by=20))

axis(2, at=seq(0, 1, by=0.1),labels=seq(0, 1, by=0.1))
```



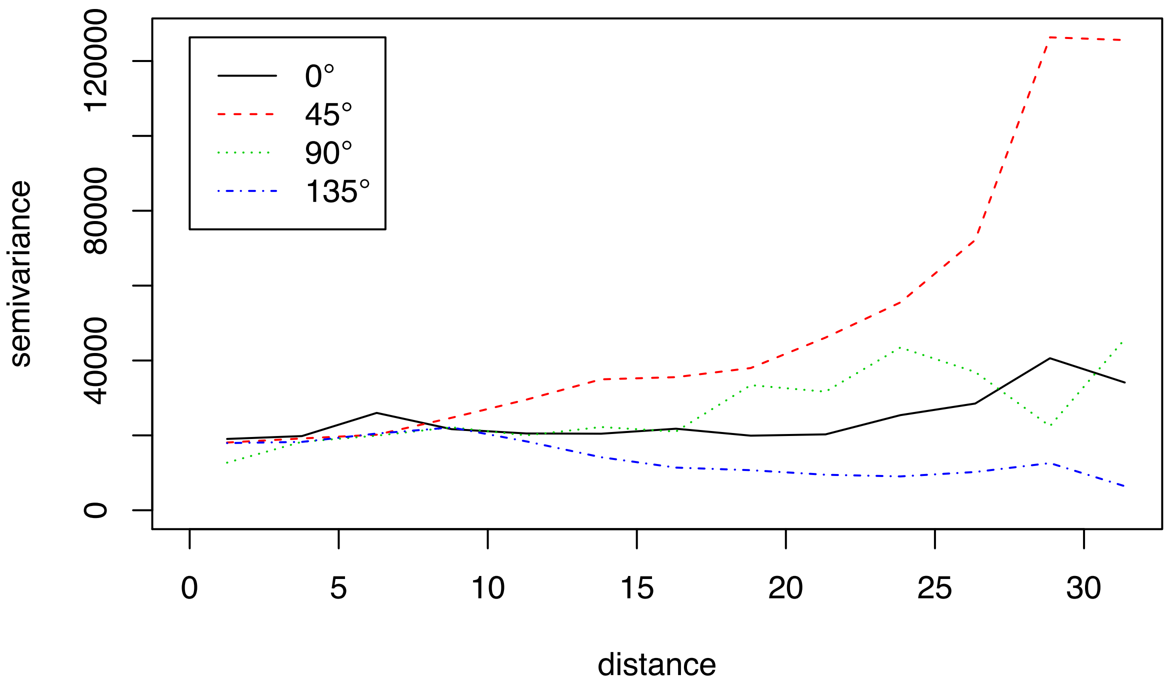

```r
b<-as.geodata(b)
plot(b)
```



```r
#plot the cariograms in for different directions
var1<-variog4(b)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 90 degrees (1.571 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 135 degrees (2.356 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing omnidirectional variogram
```

```r
plot(var1)
```




```r
#Compute the sample variogram and fit a model to it:
var1<-variog(b,dir=pi/4,estimator.type="modulus",max.dist = 24)
```

```
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
```

```r
#Plot the sample variogram:
plot(var1)

fit1<-variofit(var1,cov.model="exp",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
#Obtain the estimates of the parameters:
fit1
```

```
## variofit: model parameters estimated by WLS (weighted least squares):
## covariance model is: exponential
```

```
## parameter estimates:
##       tausq     sigmasq         phi
##    9049.553 6845895.544   13595.661
## Practical Range with cor=0.05 for asymptotic range: 40728.96
##
## variofit: minimised weighted sum of squares = 59443344747
```

```r
#Plot the model variogram:
lines(fit1)
initial.values<-expand.grid(seq(10,100,by=10),seq(10000,15000,by=200))
#npairs weights
fit1l<-variofit(var1,cov.model="exp",ini.cov.pars=initial.values,wei="npairs", fix.nugget=FALSE, nugget=
```

```
## variofit: covariance model used is exponential
## variofit: weights used: npairs
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "10"    "15000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 714925513821.107
```

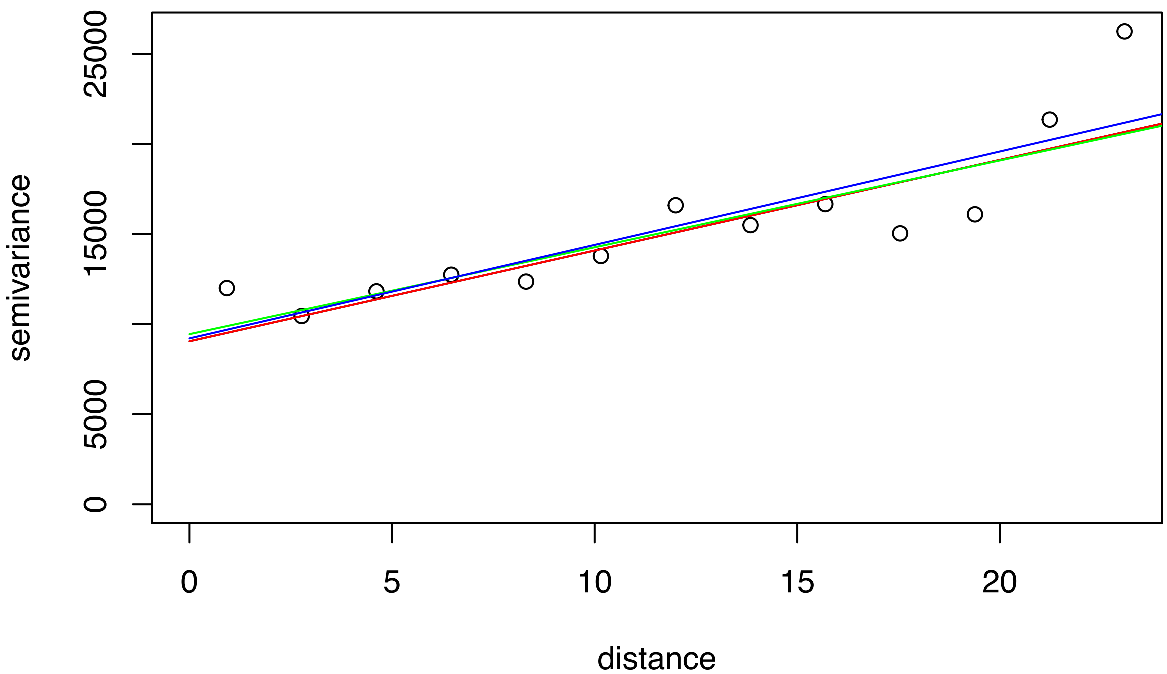
```r
lines(fit1l,lty=1,col="red")
#cressies weights
fit2l<-variofit(var1,cov.model="exp",weights="cressie",ini.cov.pars=initial.values,fix.nugget=FALSE,nugg
```

```
## variofit: covariance model used is exponential
## variofit: weights used: cressie
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "10"    "15000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 1787.31273046229
```

```r
lines(fit2l,lty=1,col="green")
#equal weights (simply OLS):
fit3l<-variofit(var1,cov.model="exp", ini.cov.pars=initial.values,weights="equal",fix.nugget=FALSE,nugge
```

```
## variofit: covariance model used is exponential
## variofit: weights used: equal
## variofit: minimisation function used: optim
## variofit: searching for best initial value ... selected values:
##               sigmasq phi     tausq   kappa
## initial.value "10"    "15000" "20000" "0.5"
## status        "est"   "est"   "est"   "fix"
## loss value: 495655426.600281
```

```r
lines(fit3l,lty=1,col="blue")
```




```
#create grid
x.range<-as.integer(range(b[[1]][,1]))
x.range
```

```
## [1] 390 411
```

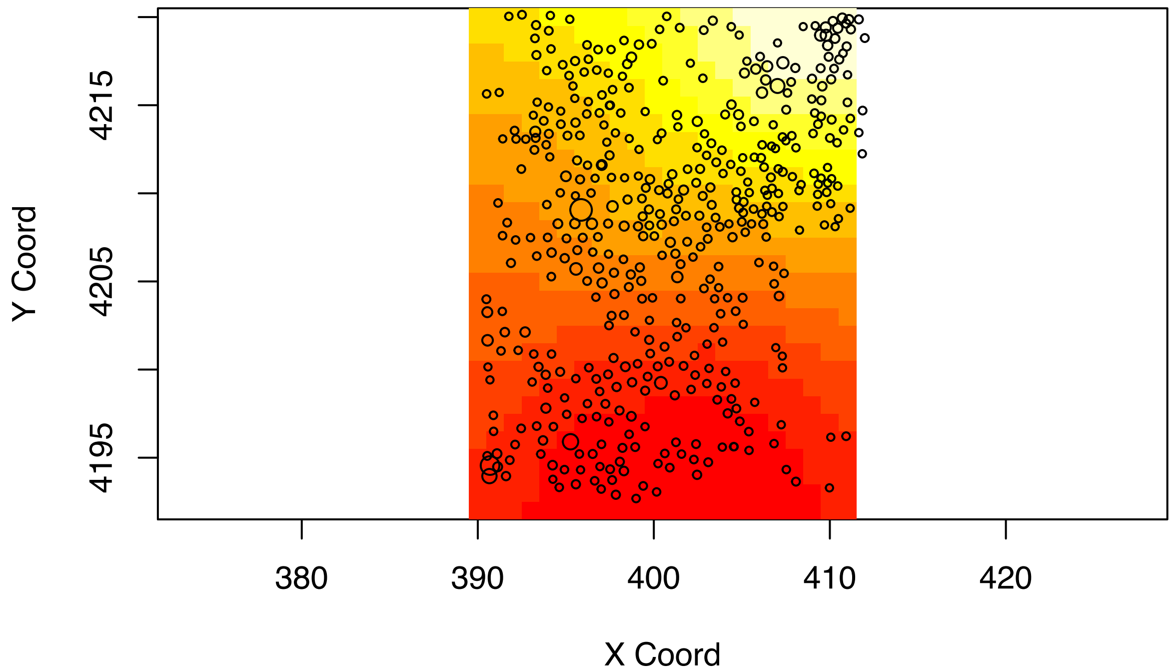
```
y.range<-as.integer(range(b[[1]][,2]))
y.range
```

```
## [1] 4192 4220
```

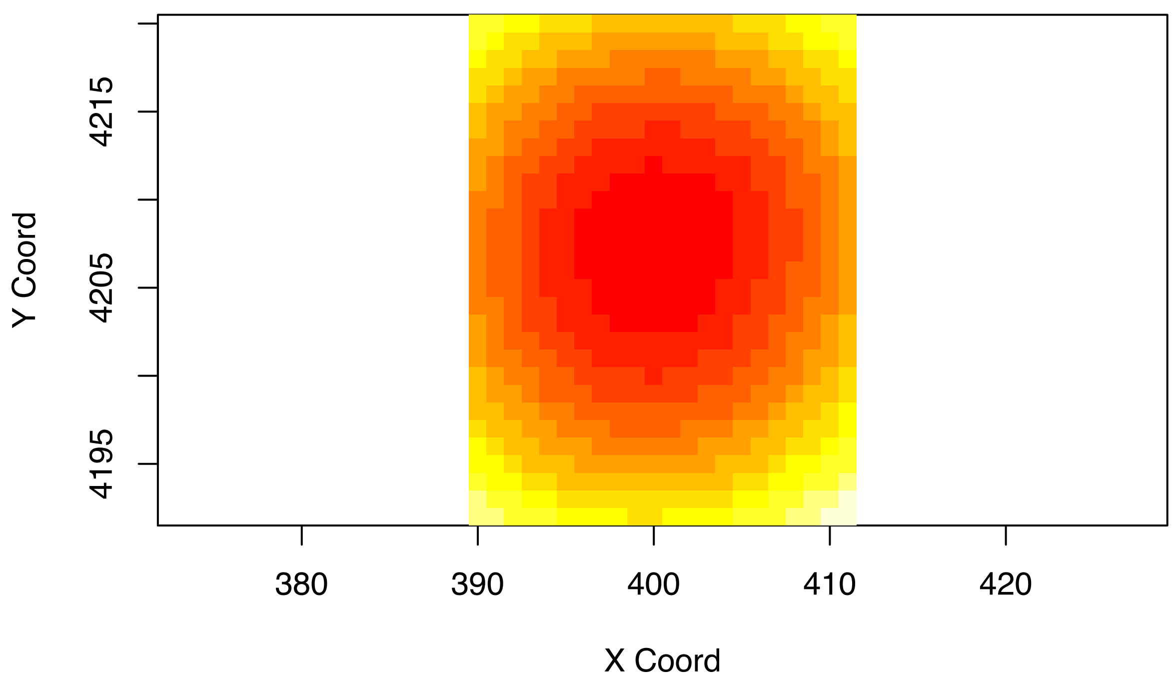
```
grd<-expand.grid(x=seq(from=x.range[1],to=x.range[2],by=1),y=seq(from=y.range[1],to=y.range[2],by=1))
library(sp)
coordinates(grd)<-~x+y
#ordinary krigging
q<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd))
```

```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```
#raster map of predict values
image(q,val=q$predict)
points(a,add=T)
```




```r
#raster map of variances
image(q,val=q$krige.var)
```

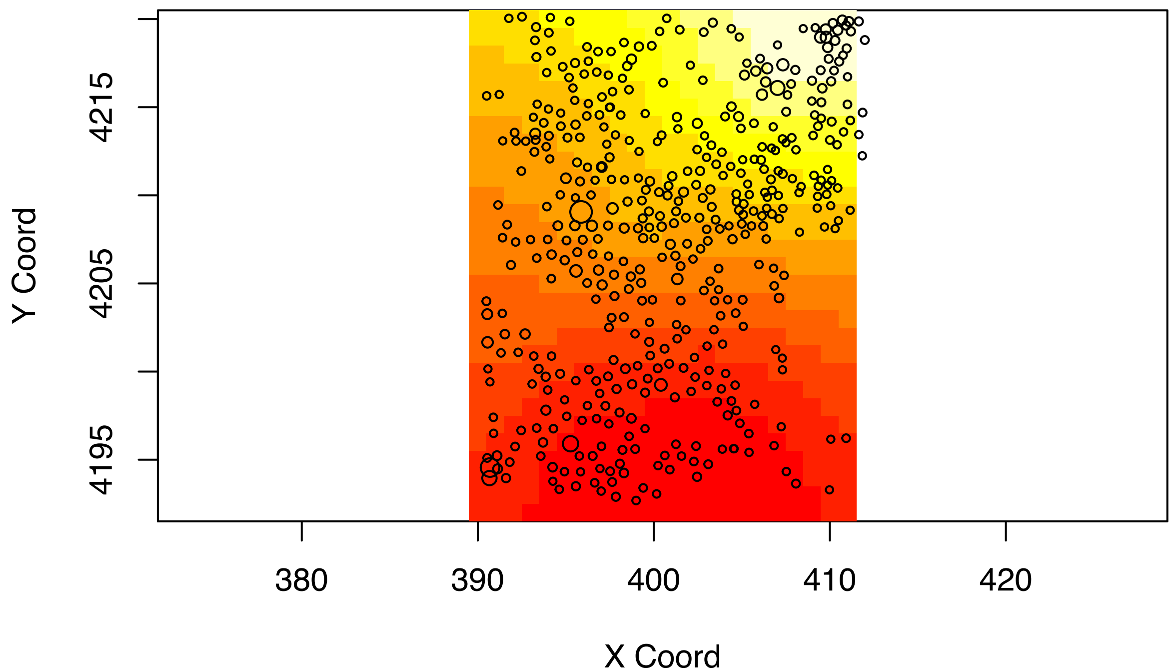



```r
#universal krigging

uq<-ksline(b,cov.model="exp",cov.pars=c(23,20000),nugget=20000,locations=data.frame(grd),trend=1)
```

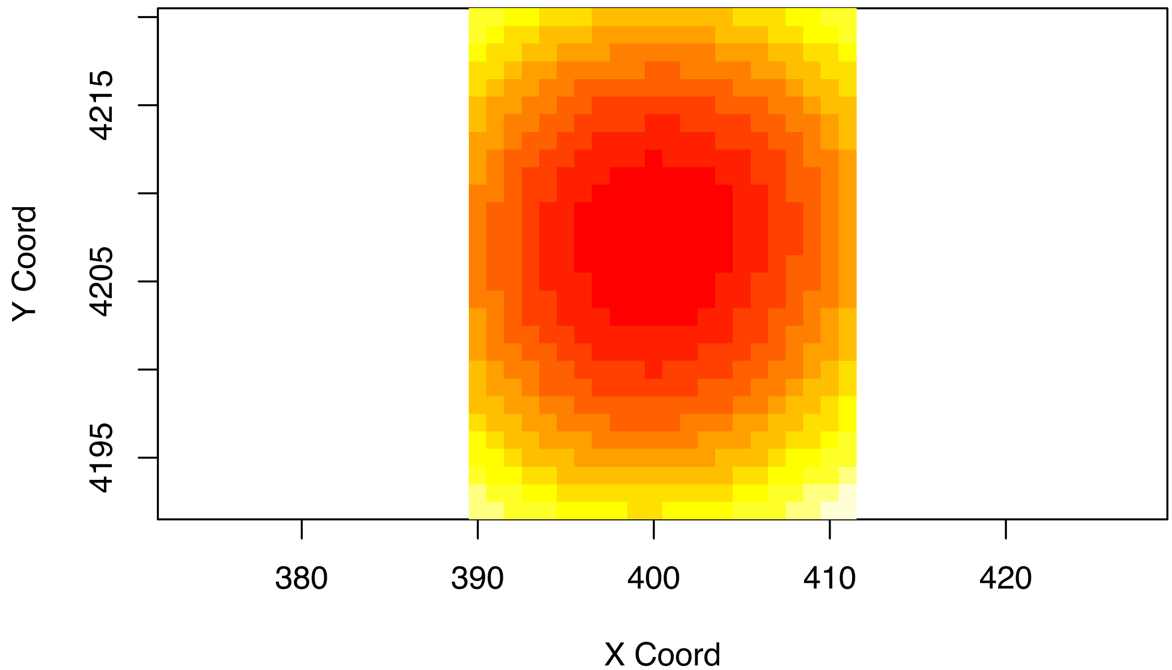
```
## ksline: kriging location:  1 out of 638
## ksline: kriging location:  101 out of 638
## ksline: kriging location:  201 out of 638
## ksline: kriging location:  301 out of 638
```

```
## ksline: kriging location:  401 out of 638
## ksline: kriging location:  501 out of 638
## ksline: kriging location:  601 out of 638
## ksline: kriging location:  638 out of 638
## Kriging performed using global neighbourhood
```

```
#raster map of predict values
image(uq,val=uq$predict)
points(a,add=T)
```




```
#raster map of variances
image(uq,val=uq$krige.var)
```



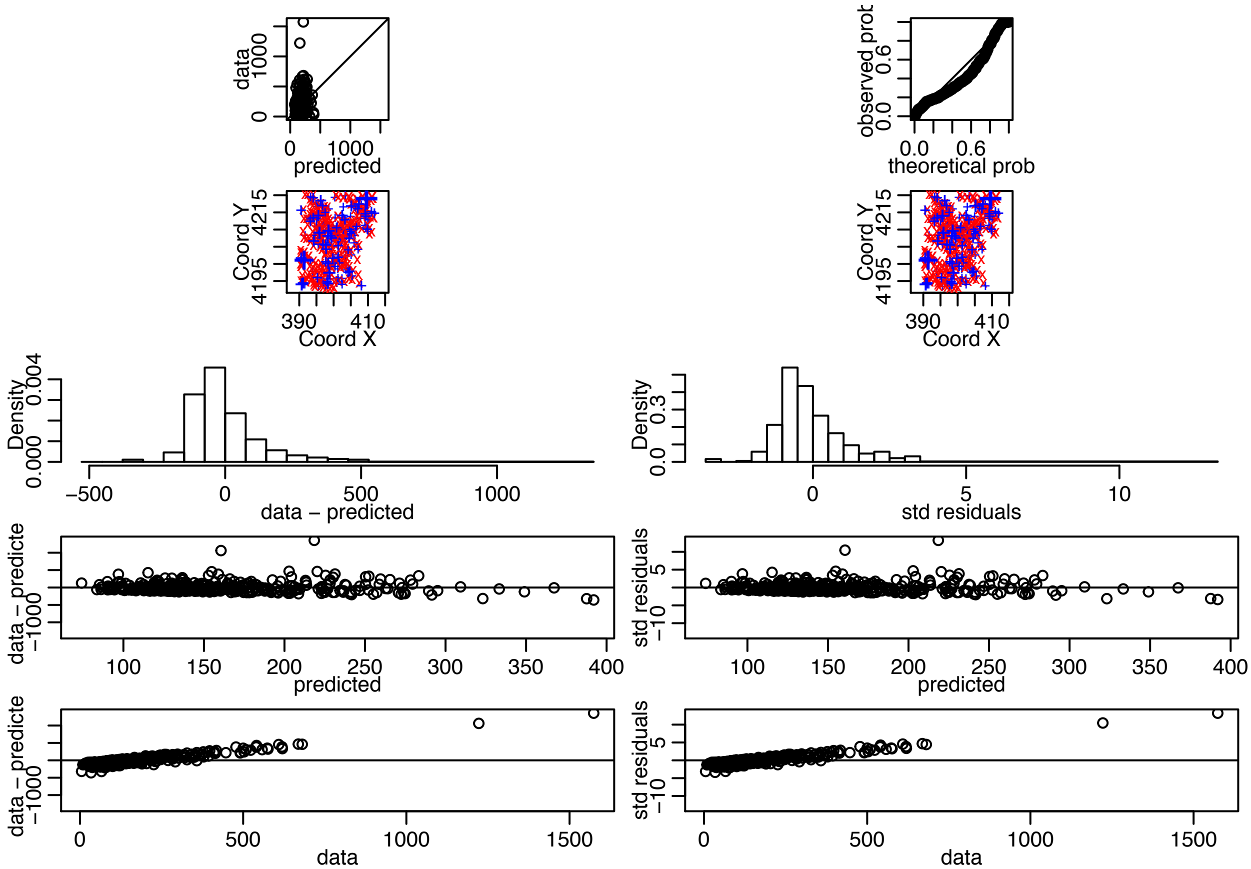

```
#Cross validation using geoR:

#Perform cross validation:
x_val1 <- xvalid(b, model=fit1)
```

```
## xvalid: number of data locations       = 377
## xvalid: number of validation locations = 377
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
## xvalid: end of cross-validation
```

```
press1 <- sum(x_val1$error^2)

press1
```

```
## [1] 8561710
```

```
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
plot(x_val1)
```




```
#Plot the sample variogram:
plot(var1)

#use sph model
fit2<-variofit(var1,cov.model="sph",ini.cov.pars=c(23,20000),max.dist=24,fix.nugget=FALSE,nugget = 2000
```

```
## variofit: covariance model used is spherical
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

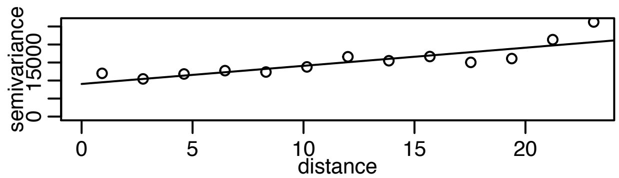
```
#Plot the model variogram:
lines(fit2)

#Perform cross validation:
x_val2 <- xvalid(b, model=fit2)
```

```
## xvalid: number of data locations       = 377
## xvalid: number of validation locations = 377
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

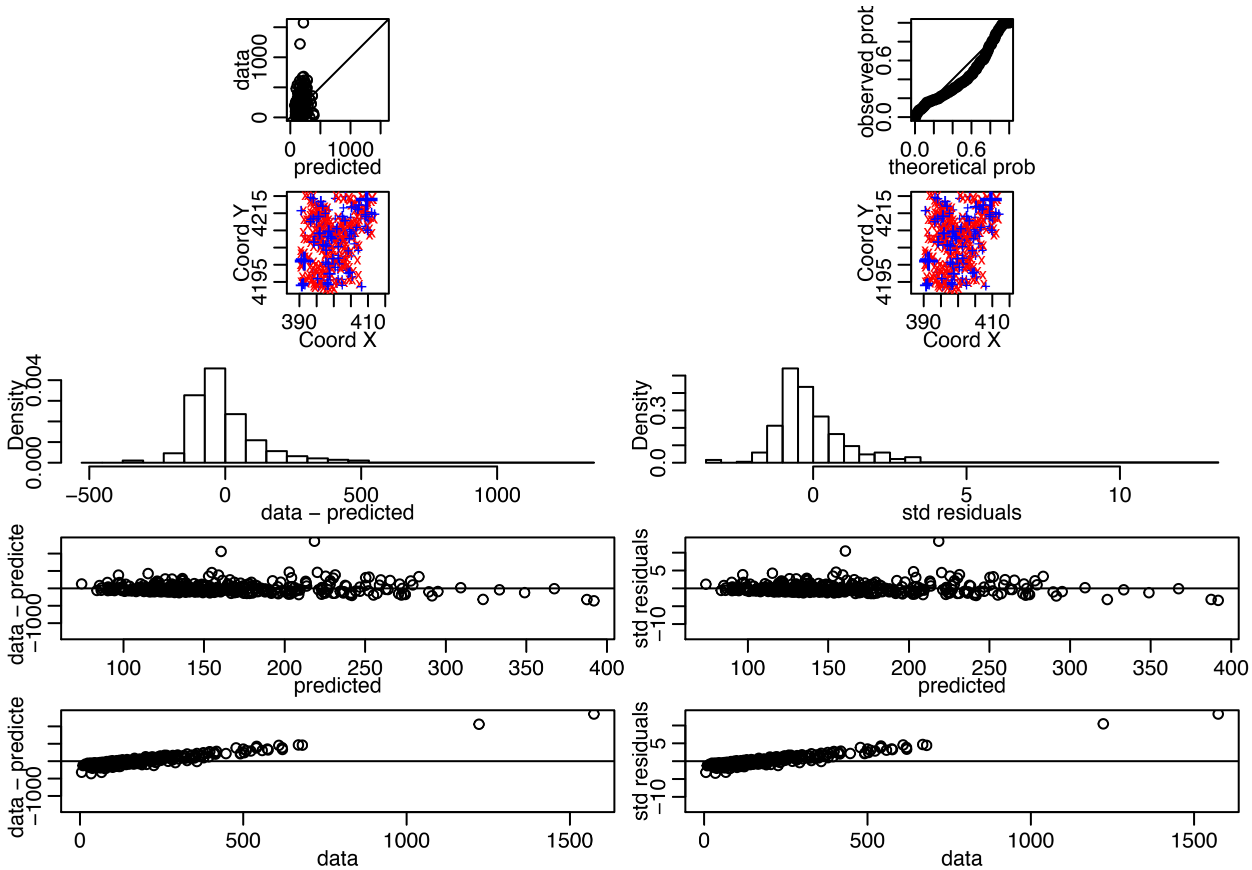
```
press2 <- sum(x_val2$error^2)

press2
```

```
## [1] 8561782
```

```
par(mfcol=c(5,2), mar=c(2.3,2.3,.5,.5), mgp=c(1.3, .6, 0))
```




```r
plot(x_val2)
```




```r
#========================================================
#Re-estimating the variogram each time a data point is omitted (exp model):
x_val_reest1 <- xvalid(b, model=fit1, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 377
## xvalid: number of validation locations = 377
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

```r
press1_reest <- sum(x_val_reest1$error^2)

press1_reest
```

```
## [1] 8590219
```

```r
#Re-estimating the variogram each time a data point is omitted (sph model):
x_val_reest2 <- xvalid(b, model=fit2, reest=TRUE, variog.obj=var1)
```

```
## xvalid: number of data locations       = 377
## xvalid: number of validation locations = 377
```

```
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
## xvalid: end of cross-validation
```

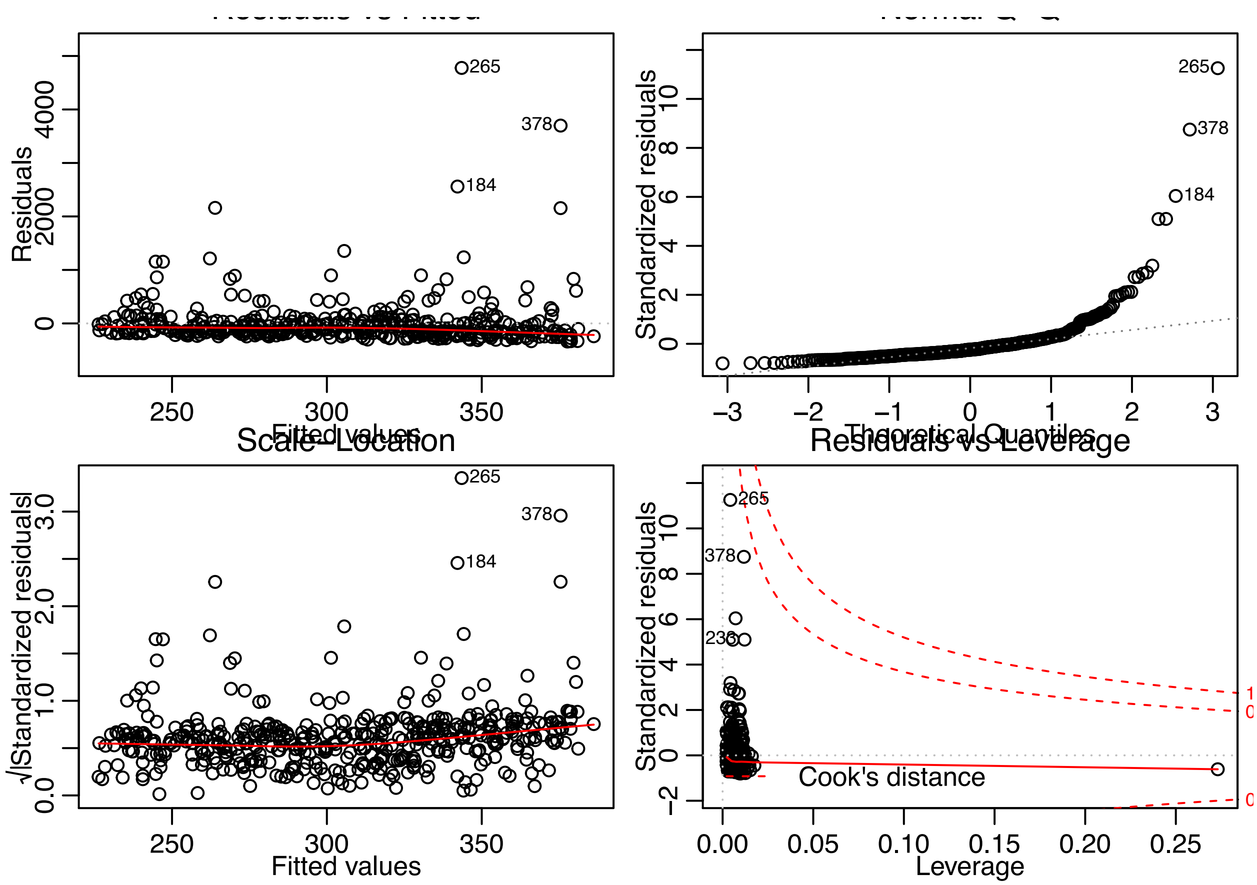
```r
press2_reest <- sum(x_val_reest2$error^2)

press2_reest
```

```
## [1] 8590284
```

```r
#Comparison of the two methods (with and without reestimation of the variogram after omitting a data po
compare <- cbind(x_val1$predicted, x_val2$predicted)
compare[1:5,]
```

```
##           [,1]      [,2]
## [1,] 194.81626 194.82294
## [2,]  91.07397  91.08279
## [3,]  91.21739  91.23270
## [4,] 162.40306 162.39261
## [5,] 129.30435 129.32923
```

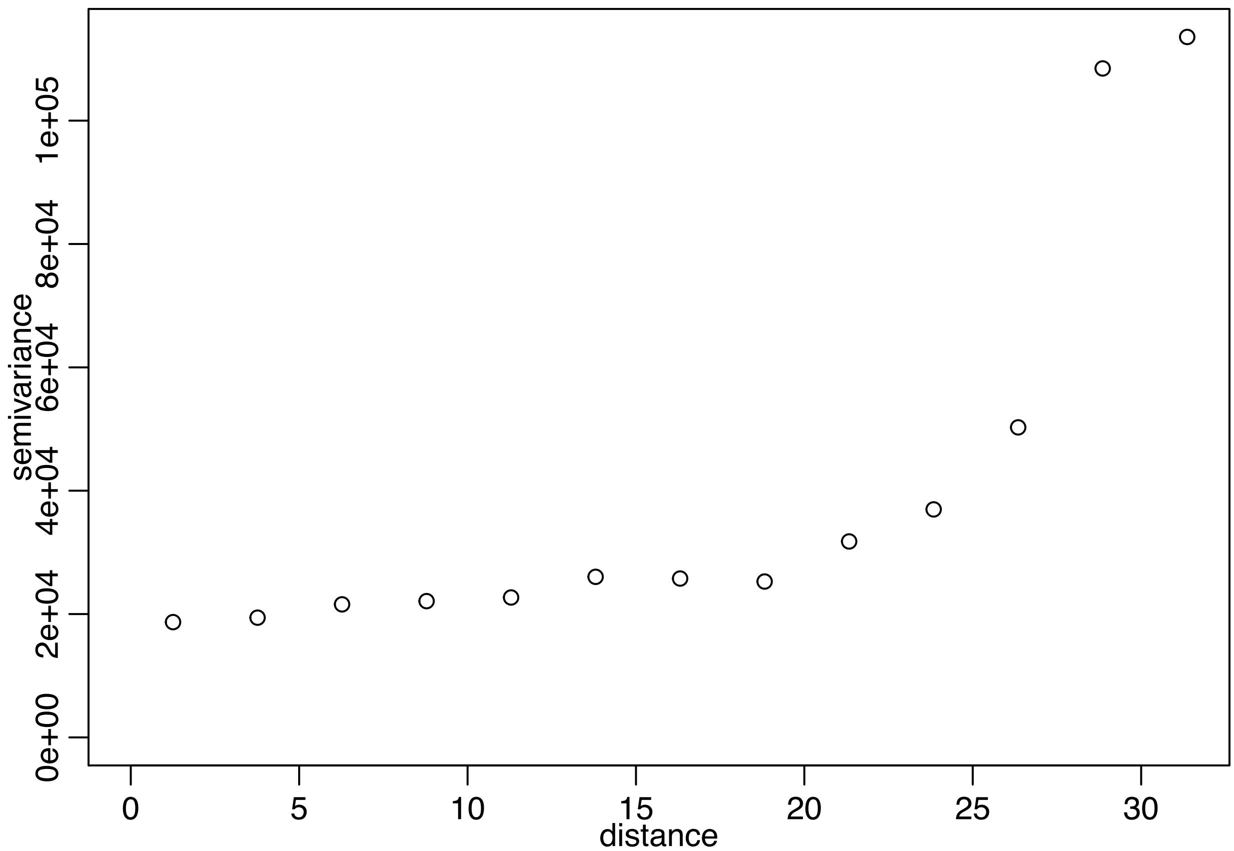
```r
#linear regression analysis
q<-lm(project$Initial_Potential ~ project$easting.Km.+project$northing.Km.)
par(mfrow=c(2,2))
plot(q)
```



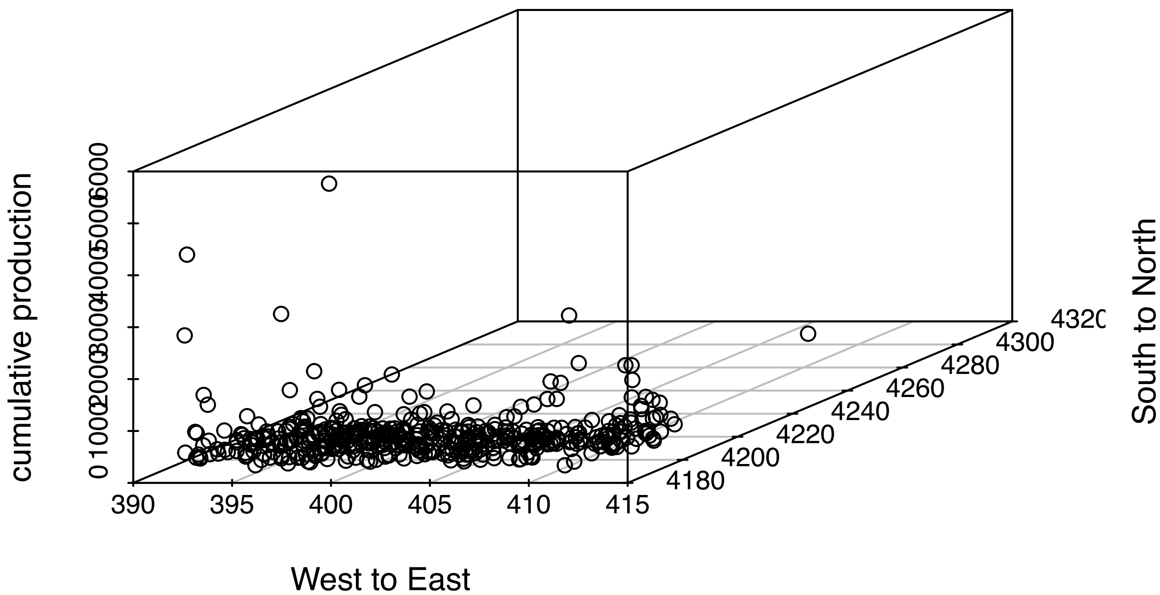

```r
#try to detrend
vario_detrend<-variog(b,trend="1st")
```

```
## variog: computing omnidirectional variogram
```

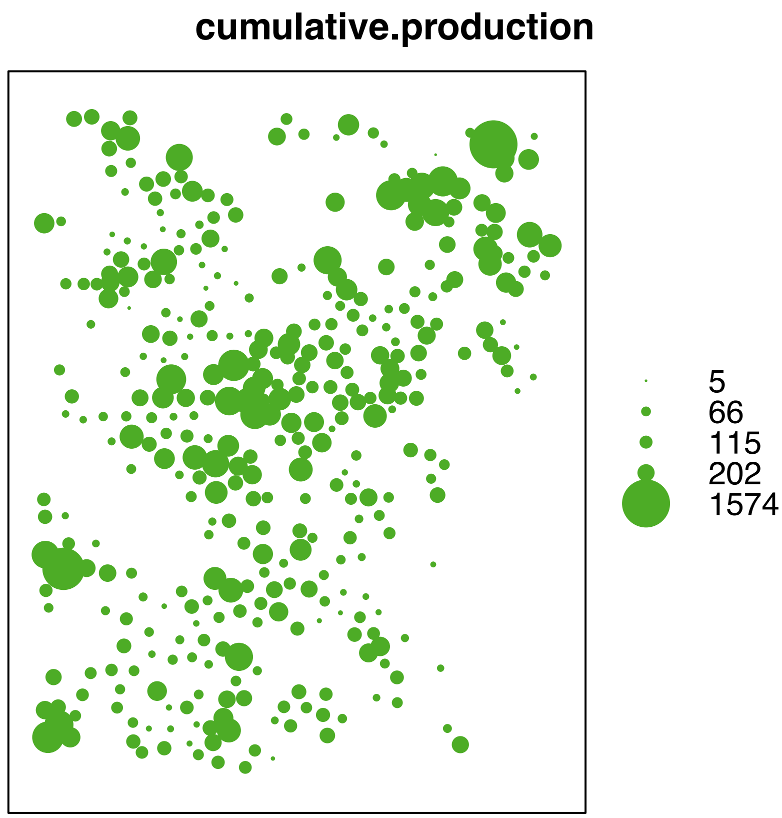
```
par(mfrow=c(1,1))
plot(vario_detrend)
```




```
#plot the 3d data points
library(scatterplot3d)
scatterplot3d(project$easting.Km.,project$northing.Km.,project$Initial_Potential,xlab="West to East",yla
```



```
#bubble plots
library(gstat)
library(sp)
b<-data.frame(b)
colnames(b)<-c("x","y","cumulative.production")
coordinates(b)<- ~x+y
bubble(b,"cumulative.production")
```



We can see there is no big difference btween press1 and press2. Press1 is slightly smaller than press 2.