Week 5: Geostatistics II

More Notes on Assumption of Kriging

Stationarity

Consider a spatial process $Z(s)$ with a mean $m(s)$ and variance $\sigma^2(s)$ exists $\forall s \in \mathcal{D}$.

1. The process is `strictly stationary` or `strongly stationary` if, for any given $n \geq 1$, any set set of n sites and any $h \in \mathcal{R}^d$, the distribution of $Z(s_i), \dots, Z(s_n)$ is the same as $Z(s_i + h), \dots, Z(s_n + h)$
2. A less restrictive assumption is `weak stationarity` or `second-order stationarity`, which is to assume $m(s) \equiv \mu$ and $cov[Z(s_i), Z(s_i + h)] = C(h)$ for any $h \in \mathcal{R}^d$ s.t. both s_i and $s_i + h$ are within $\mathcal{D}$. $C(h)$ is `covariogram`.
    - $cov[Z(s_i), Z(s_i + h)] = E[Z(s_i) - \mu][Z(s_i + h) - \mu] = E[Z(s_i)Z(s_i + h)] - \mu^2 = C(h)$
    - $\sigma^2(Z(s_i)) = E[Z(s_i) - \mu]^2 = E[Z(s_i)^2] - \mu^2 = C(0)$
    - $\rho(h) = \frac{C(h)}{\sigma(Z(s_i))\sigma(Z(s_i+h))}$ is `correlogram`
3. The `second-order stationarity` assumes the existence of covariance. For cases where covariance and variance do not exist, we assume the stationarity of the difference.
    - $E[Z(s)] = \mu, \forall s$
    - $\sigma^2[Z(s_i + h) - Z(s_i)] = E[Z(s_i + h) - Z(s_i)]^2 = 2\gamma(h)$
    - $\hat{\gamma(h)} = \frac{1}{2N(h)} \sum_{(s_i,s_j)\in N(h)}^{N(h)} [Z(s_i) - Z(s_j)]^2$
    - $2\gamma(h)$ is called `variogram` and $2\gamma(h)$ is therefore `semivariogram`
    - `intrinsic stationarity`

Variogram

Not every function could be used as a variogram. Suppose there are stationary random variables $Z(s_1), Z(s_2), \dots, Z(s_n)$ with expectation μ and covariance $C(h)$. Let Y be the linear combination of these variables. $Y = \sum_{i=1}^{n} \omega_i Z(s_i)$, which is also a random variable itself and must have nonnegative variance.

$$\sigma^2(Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C(s_i - s_j) \geq 0$$

Therefore legal covariance functions much ensure that the variance of Y is always non-negative. This type of functions is often referred to as `non-negative definite`. If $\sigma(Y) \geq 0$, it is called `positive definite`.

Since for variogram $\gamma(h) = C(0) - C(h)$. The variance of Y, $\sigma^2(Y)$, can be rewritten using $\gamma(h)$:

$$\sigma^2(Y) = C(0) \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j - \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j \gamma(s_i - s_j) \geq 0$$

In case of $\sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j = 0$:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j \gamma(s_i - s_j) \leq 0$$

Therefore, legal variogram function must be `negative definite`.

The common examples of the legal variogram or covariogram functions include `Spherical`, `Exponential`, `Gaussian`, `Matern` that we have used for the last week.

Kriging

Spatial prediction are commonly represented as:

$$Y(s) = m(s) + Z(s) + \epsilon$$

,

where $Y(s)$ represent the primary variable of study at location s, $m(s) = \beta X(s)$ is the mean or trend component (or trend) that could be modeled through covariate $X(s)$, $Z(s)$ is the spatial effect that are often assumed as Gaussian distributed specified by covariance functions $C(h; \phi, \sigma^2)$, ϵ is the random noises (nugget effect) specified by parameter τ^2

Different specifications of the trend component $m(s)$ leads to different types of kriging methods.

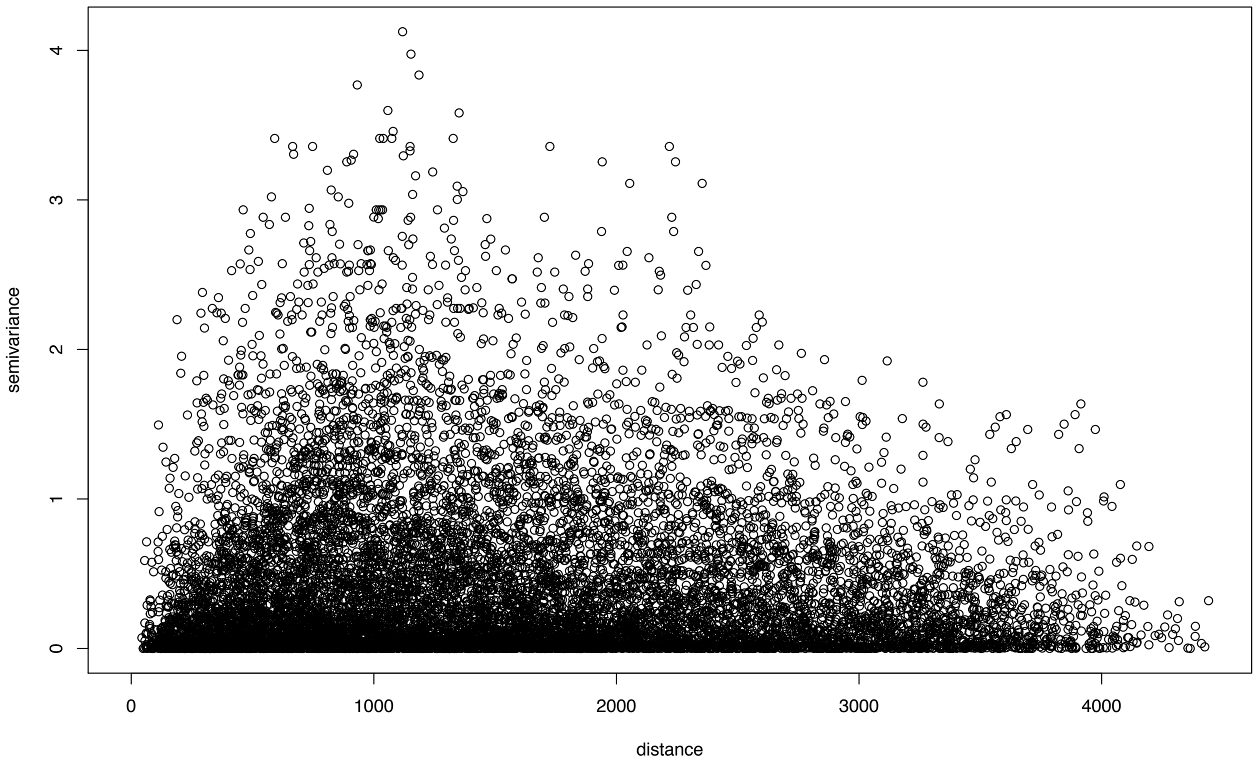
- Simple kriging: $m(s) \equiv \mu$
- Ordinary kriging: $m(s)$ is constant but unknown
- Regression kriging: $m(s)$ is modeled through covariates $X(s)$
  - Universal kriging
  - Kriging with external drift

Under the assumption of Gaussian distributed variables, kriging leads to the unbiased linear estimation with minimum variance or best linear unbiased estimation (BLUE).

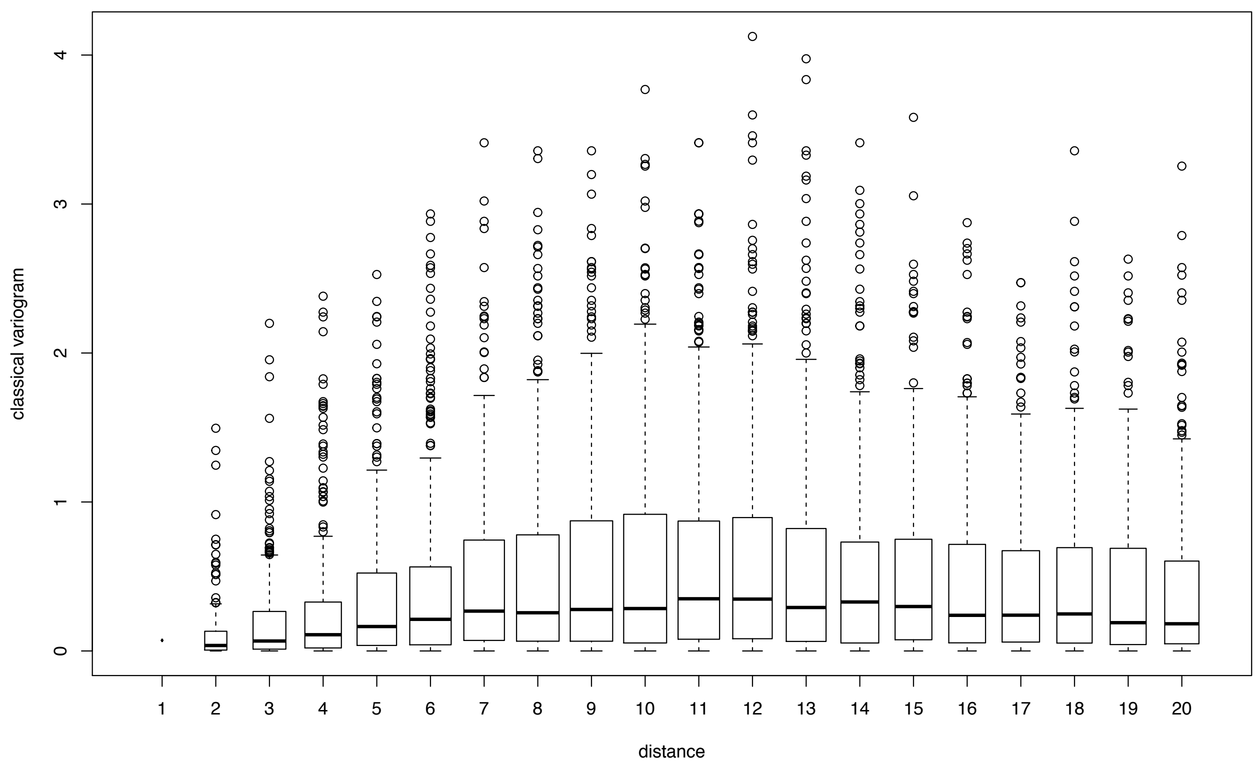
If the measurements cannot satisfy the Gaussian assumption, transformation usually needs to be performed first (e.g., log transform) or Bayesian hierarchical model can be used.

Excises: Model-based Geostatistics

```
data(meuse)
meuse=cbind(meuse, log(meuse$lead))
# convert it to a geodata object that geoR requires.
meuse=as.geodata(meuse,coords.col=1:2, data.col=15, covar.col=8)
# generate variogram cloud. geoR provides two different ways for the sample
# variogram values, classical and modules. The classical one is the one we
# talked about in the class, and the modules one is the
cloud1 <- variog(meuse, option = "cloud", estimation.type='classical')
names(cloud1)
head(cloud1$u, n=20)
head(cloud1$v, n=20)
plot(cloud1)
```

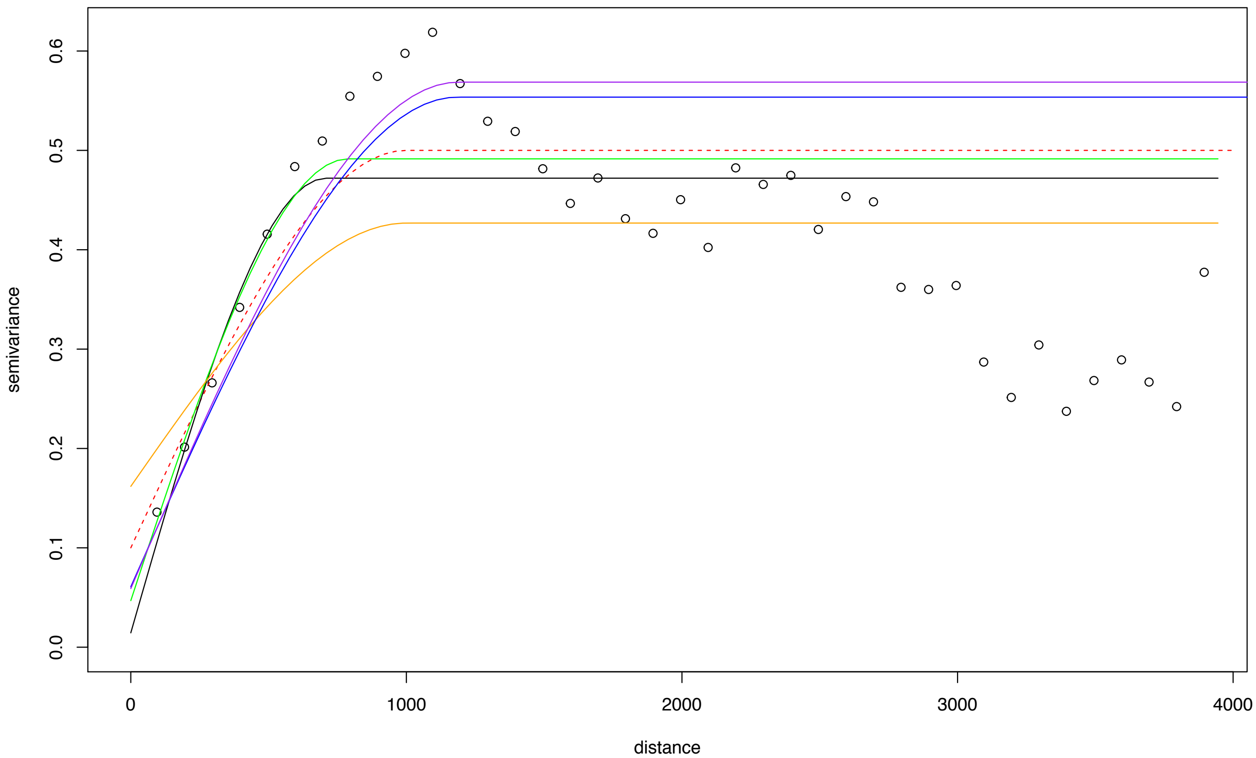



```
#box-plot of the cloud
bin1 <- variog(meuse, breaks=seq(45, 2000, by = 100), estimation.type='classical',bin.cloud=T, max.dista
plot(bin1, bin.cloud=T)
```

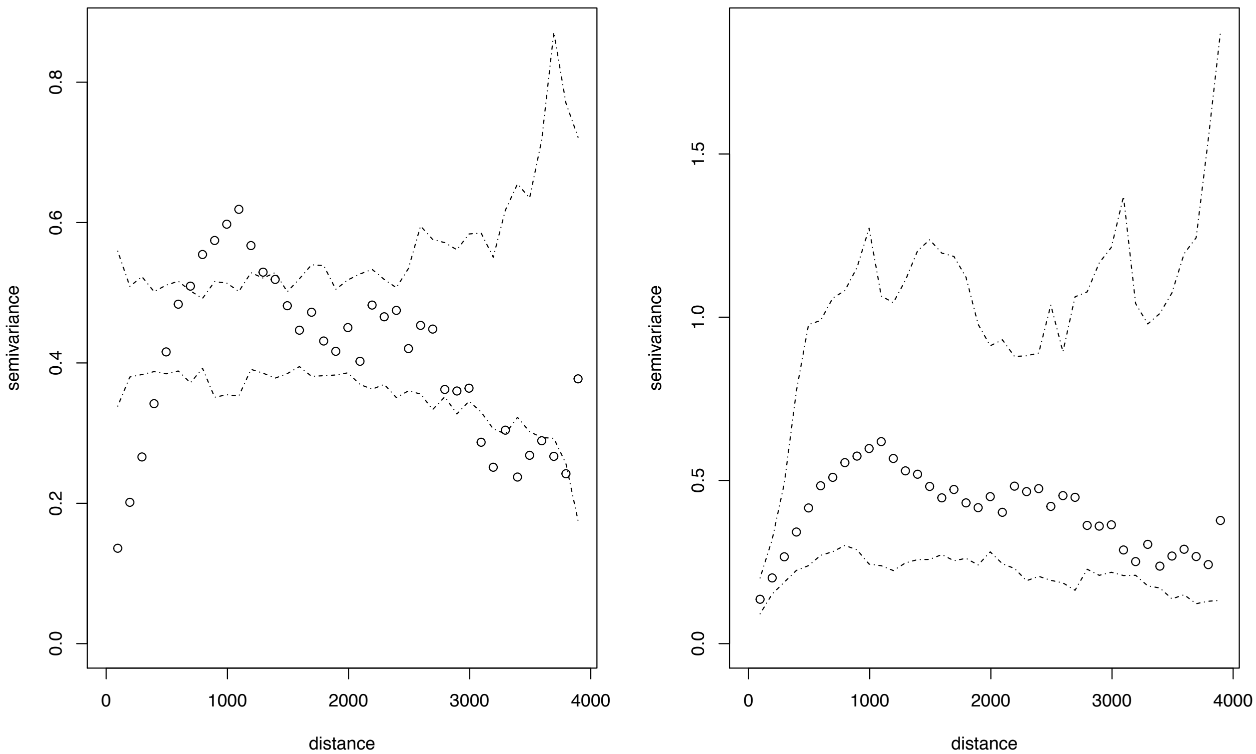



```
#variogram fitting
```

```r
##by eye
variogram <- variog(meuse, breaks=seq(45, 4000, by = 100))
plot(variogram)
lines.variomodel(cov.model="sph", cov.pars=c(0.4,1000), nug=0.1, max.dist=4000, lty=2, col='red')

#Fit the spherical variogram using the default option (check ?variofit manual).
fit1 <- variofit(variogram, cov.model="sph", ini.cov.pars=c(0.4,1000), fix.nugget=FALSE, nugget=0.1)

lines(fit1, lty=1)

#Use Cressies weights:
fit2 <- variofit(variogram, cov.model="sph", weights="cressie", ini.cov.pars=c(0.4,1000), fix.nugget=FA

lines(fit2, lty=1, col="green")

#Use equal weights (simply OLS):
fit3 <- variofit(variogram, cov.model="sph", ini.cov.pars=c(0.4,1000), weights="equal", fix.nugget=FALS

lines(fit3, lty=1, col="orange")

#MML:
ml <- likfit(meuse, cov.model="sph", ini.cov.pars=c(0.4,1000), fix.nugget=FALSE, nugget=0.1)

lines(ml, col="blue")

#REML:
rml <- likfit(meuse, cov.model="sph", ini.cov.pars=c(0.4,1000), fix.nugget=FALSE, nugget=0.1, lik.metho

lines(rml, col="purple")
```



```r
#
env.mc <- variog.mc.env(meuse, obj.var=variogram)
env.model <- variog.model.env(meuse, obj.var=variogram, model=fit2)
par(mfrow=c(1,2))
plot(variogram, envelope=env.mc)
plot(variogram, envelope=env.model)
```

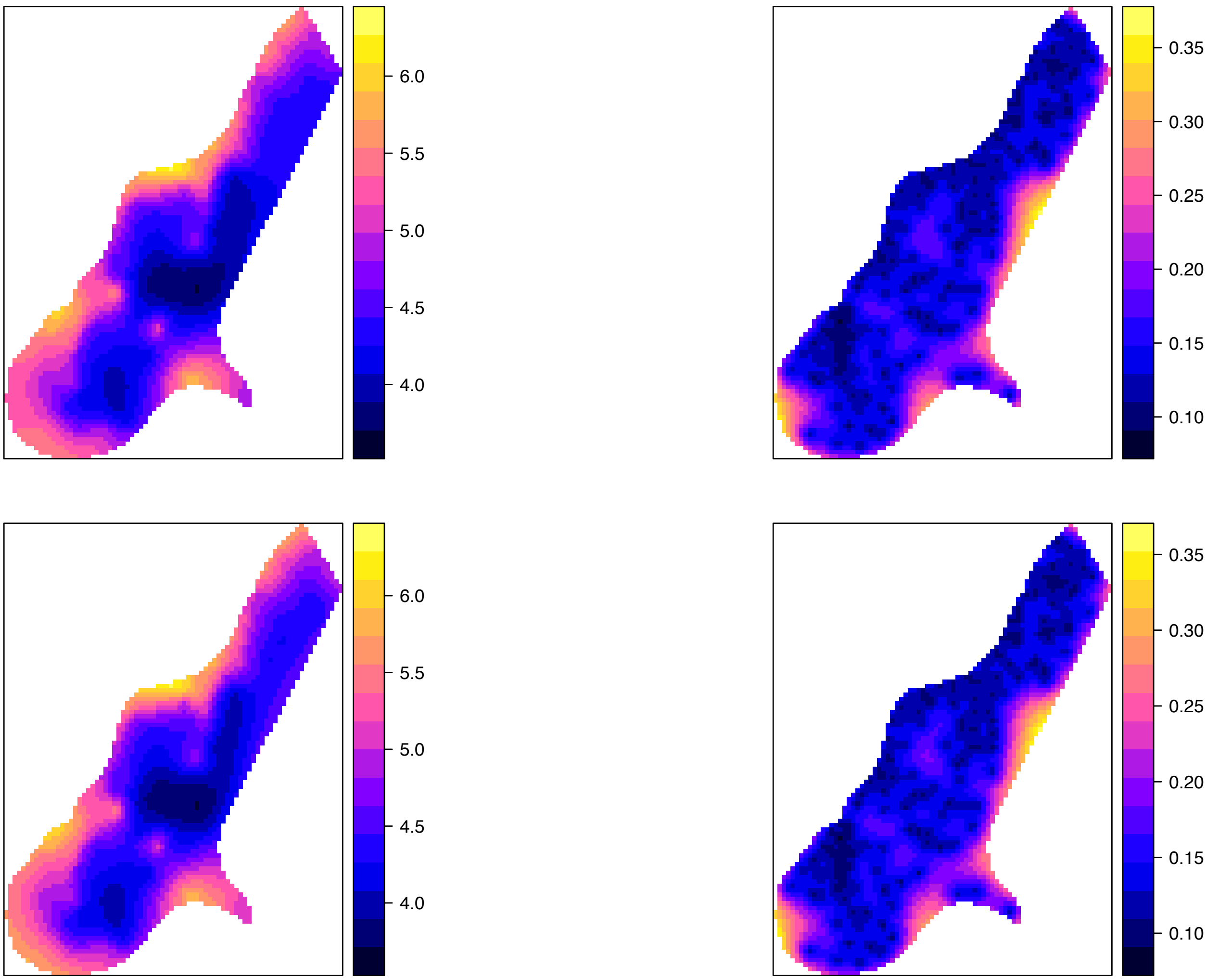



```r
# profile likelihood
#prof <- proflik(ml, geodata = meuse, sill.val = seq(0.20, 1, l = 5), range.val = seq(800, 1200, l = 6)
#plot(prof)
```

Simple Kriging and Ordinary Kriging

```r
data(meuse.grid)
predSites=cbind(meuse.grid$x, meuse.grid$y)
OKpred=krige.conv(meuse, locations=predSites, krige=krige.control(type.krige='ok', obj.m=ml))

OKresult=as.data.frame(cbind(OKpred$predict, OKpred$krige.var))
coordinates(OKresult)=predSites
gridded(OKresult)=TRUE

# the constant mean is specified by beta
mean(meuse$data)
SKpred=krige.conv(meuse, locations=predSites, krige=krige.control(type.krige='sk', obj.m=ml, beta=6))
SKresult=as.data.frame(cbind(SKpred$predict, SKpred$krige.var))
coordinates(SKresult)=predSites
gridded(SKresult)=TRUE
```

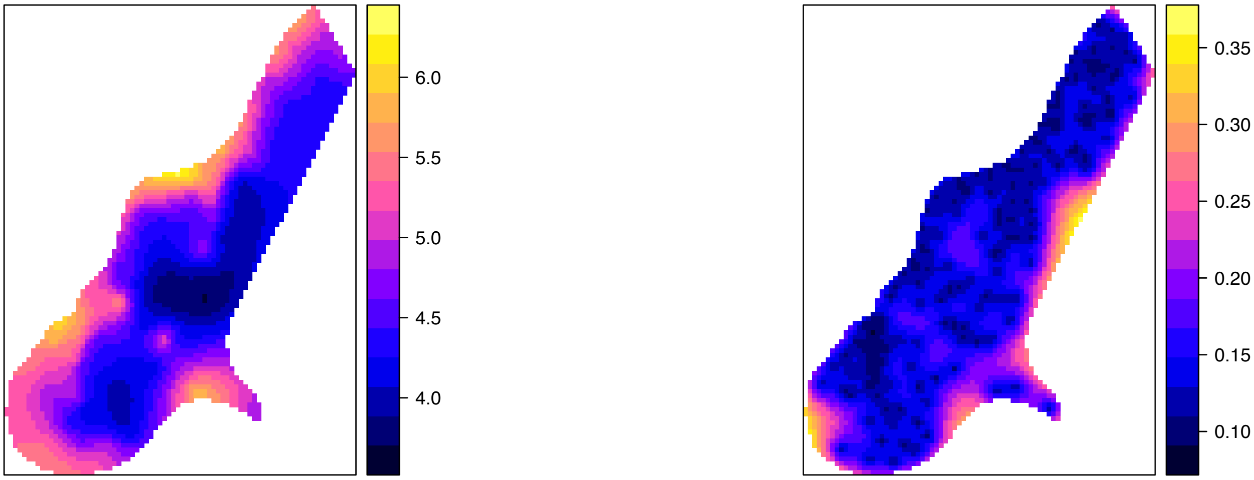
```r
p1=spplot ( OKresult, "V1")
p2=spplot(OKresult,"V2")
p3=spplot(SKresult,"V1")
p4=spplot(SKresult,"V2")

print(p1, position = c(0,.5,.5,1),more=T)
print(p2, position = c(.5,.5,1,1),more = T)
print(p3, position = c(0,0,.5,0.5),more=T)
print(p4, position = c(.5,0,1,0.5))
```



Regression Kriging

```r
RKpred=krige.conv(meuse, locations=predSites, krige=krige.control(type.krige='ok', obj.m=ml, trend.d=~d
RKresult=as.data.frame(cbind(RKpred$predict, RKpred$krige.var))
coordinates(RKresult)=predSites
gridded(RKresult)=TRUE

p5=spplot ( RKresult, "V1")
```

```r
p6=spplot(RKresult,"V2")

print(p5, position = c(0,.5,.5,1),more=T)
print(p6, position = c(.5,.5,1,1))
```



Bayes Kriging

```r
#Warining: the following codes are very computationally demanding. In the
#interest of time,  I made the specification of the model unrealistically
# simple.

xrange=range(meuse$coords[,1])
yrange=range(meuse$coords[,2])
x=seq(xrange[1], xrange[2], length=10)
y=seq(yrange[1], yrange[2], length=10)
predSites=expand.grid(x,y)

#model.spec <- model.control(trend.d=~dist, trend.l=~meuse.grid$dist, cov.model="matern", kappa=0.5, la

model.spec <- model.control(cov.model="matern", kappa=0.5, lambda=1)

prior.spec <- prior.control(beta.prior="flat",sigmasq.prior="reciprocal",tausq.rel.prior="uniform",taus

output.spec <- output.control(quantile=c(0.50,0.025,0.975), n.post=100, n.pred=100)

bayes1 <- krige.bayes(meuse, locations=predSites, borders=NULL, model=model.spec, prior=prior.spec, outp

out <- bayes1$posterior
out <- out$sample
beta0.qnt <- quantile(out$beta0, c(0.50,0.025,0.975))
beta1.qnt <- quantile(out$beta1, c(0.50,0.025,0.975))
phi.qnt <- quantile(out$phi, c(0.50,0.025,0.975))
sigmasq.qnt <- quantile(out$sigmasq, c(0.50,0.025,0.975))
tausq.rel.qnt <- quantile(out$tausq.rel, c(0.50,0.025,0.975))
tausq <- (out$tausq.rel)*(out$sigmasq)
tausq.qnt <- quantile(tausq, c(0.50,0.025,0.975))
```

```r
samples.lead<- cbind(out$beta0,out$beta1,out$phi,out$sigmasq,tausq)
summary.lead <- rbind(beta0.qnt,beta1.qnt,phi.qnt,sigmasq.qnt,tausq.qnt)

out2 <- bayes1$predictive
predictive.mean <- out2$mean.simulations
predictive.variance <- out2$variance.simulations
predictive.sd <- sqrt(predictive.variance)
predictive.quantiles <- out2$quantiles.simulations


summary.predictive <- cbind(predictive.mean,predictive.sd,predictive.quantiles)
```