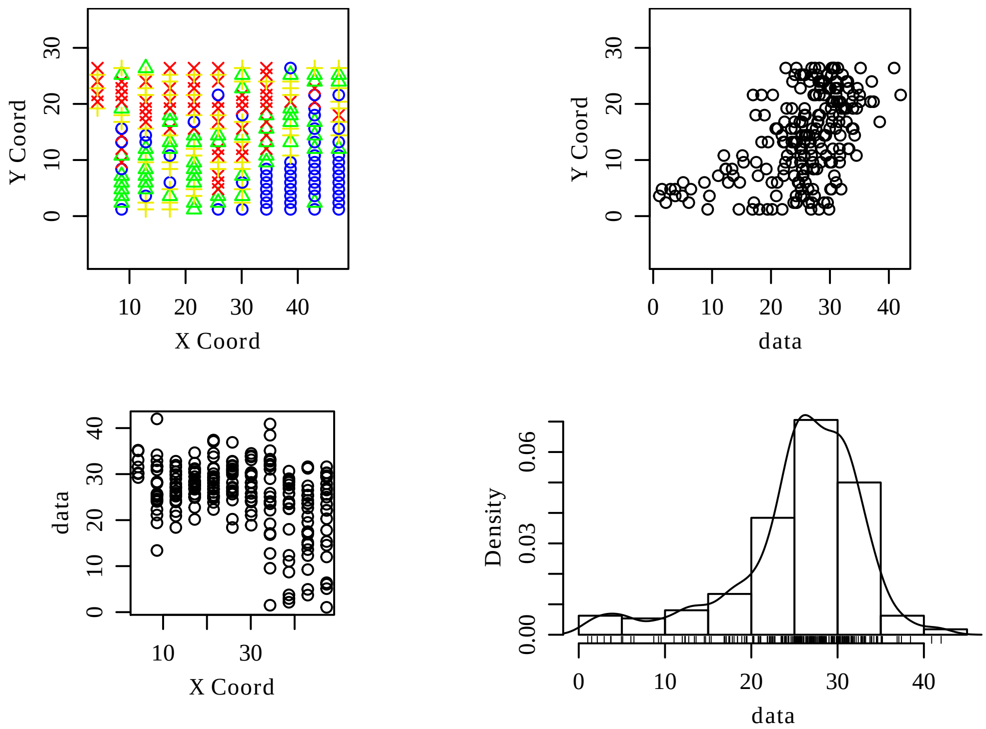
1. *A data set (wheat.txt) has been sent to you. The data set contains yields of wheat recorded at spatial coordinates. Note that the header is x, y, and z with z being the yields. We will need a couple of different data object types. Pay attention to the R code below. Do not worry about anisotropy.*

```
wt <- read.table("wheat.txt", header = TRUE)
require(geoR)
head(wt)

    x    y     z
1 4.3 19.2 29.25
2 4.3 20.4 31.55
3 4.3 21.6 35.05
4 4.3 22.8 30.10
5 4.3 24.0 33.05
6 4.3 25.2 30.25

summary(wt)

       x               y               z
 Min.   : 4.30   Min.   : 1.20   Min.   : 1.05
 1st Qu.:17.20   1st Qu.: 7.20   1st Qu.:23.52
 Median :25.80   Median :14.40   Median :26.85
 Mean   :27.22   Mean   :14.08   Mean   :25.53
 3rd Qu.:38.70   3rd Qu.:20.40   3rd Qu.:30.39
 Max.   :47.30   Max.   :26.40   Max.   :42.00

str(wt)

'data.frame': 224 obs. of  3 variables:
 $ x: num  4.3 4.3 4.3 4.3 4.3 4.3 4.3 8.6 8.6 8.6 ...
 $ y: num  19.2 20.4 21.6 22.8 24 25.2 26.4 1.2 2.4 3.6 ...
 $ z: num  29.2 31.6 35 30.1 33 ...

wt <- data.frame(apply(wt,2,function(t){
  as.numeric(as.character(t))
}))

wheat.dat <- wt
wheat.geodat <- as.geodata(wheat.dat, coords.col=1:2,data.col=3)

# warnings for eliminating rows with NAs

wheat.grid <- expand.grid(seq(0,50,l=25),
                           seq(0,30,l=25))
```

   (a) *Plot the data and comment on the results.*

   There appears to be a trend in both the east-west and north-south directions. The data are skewed left. Quantiles associated with the blue color are clustered mostly at high X coordinate values and low Y coordinate values.

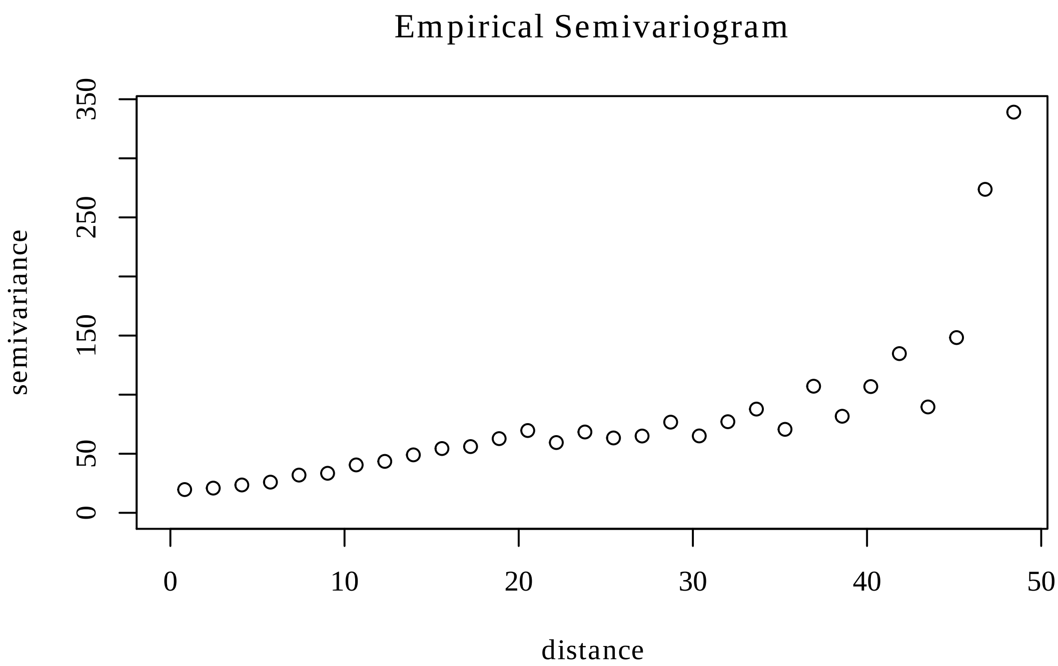
```
plot(wheat.geodat)
```



(b) *Produce a plot of the empirical semivariogram of the wheat yields. Can this plot be trusted for estimation of semivariogram parameters to be used in kriging. Why or why not?*

In the presence of a trend, which we saw in (a), the empirical semivariogram estimator is biased.

```
wheat.variog <- variog(wheat.geodat, max.dist = max(dist(wheat.geodat$coords)),
                       uvec = 30, messages = FALSE)

plot(wheat.variog, main = "Empirical Semivariogram")
```

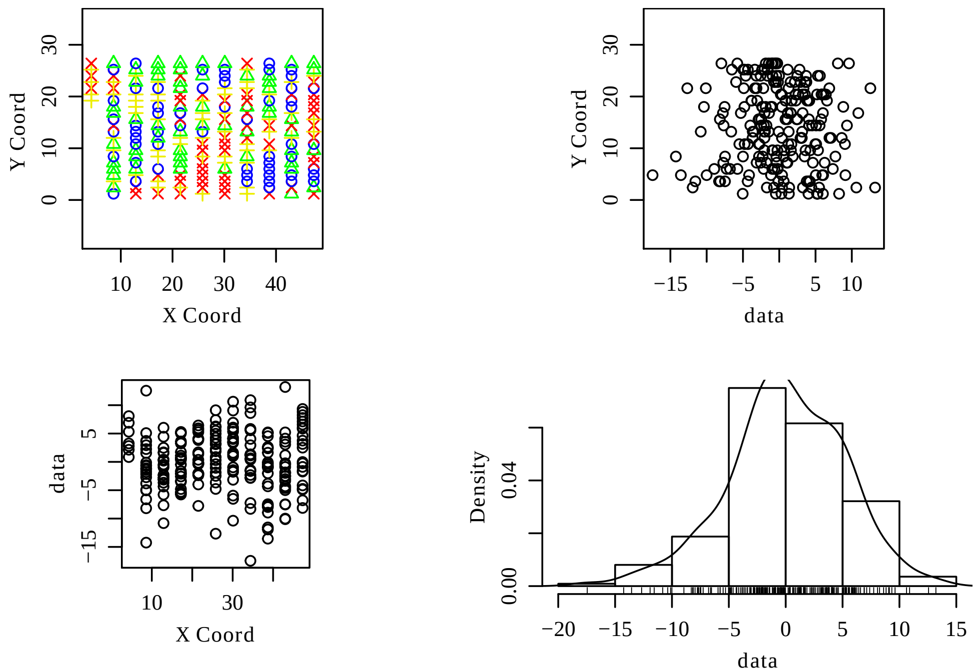


(c) *We will use the surf.ls function in the spatial library to fit a quadratic trend model to the yields by ordinary least squares and plot the empirical semivariogram of the residuals. Fit an appropriate semivariogram model to the semivariogram using your method of choice. Justify your final selection.*

I used the REML method because theoretically it provides valid standard errors if the assumptions are met (p. 24) and REML ensures positive variance estimates. I used initial covariance parameters estimated "by eye" from the empirical semi-variogram.

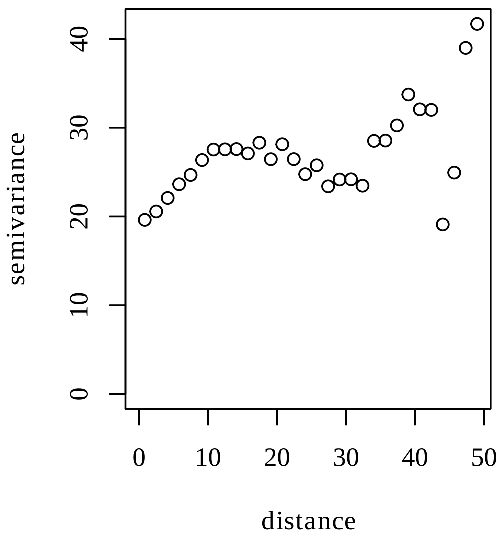
I considered all the exponential, spherical, and gaussian models. The final model I will choose is the spherical model because the corresponding estimates are much more reasonable than either the exponential or gaussian model estimates.

```
require(spatial)
wheat.ls<-surf.ls(2,wheat.dat) # fits a second order polynomial trend surface
resid.dat<-cbind(wheat.dat$x,wheat.dat$y,residuals(wheat.ls))
resid.geodat<-as.geodata(resid.dat,coords.col=1:2,data.col=3)

plot(resid.geodat)
```




```
par(mfrow=c(1,2))

#no outliers, use classical
resid.class <- variog(resid.geodat, coords = resid.geodat$coords, data = resid.geodat$data, uvec = 30,
            estimator.type = "classical", messages = FALSE)
plot(resid.class)
#not attaining sill

#cov.pars=(partialsill= 42-20, phi = 50)
#cov.model = "exponential"
#fix.nugget = TRUE -- it is quite stable?
#nugget = 20

exp.ml <- likfit(resid.geodat, ini=c(22,50/3), fix.nugget = TRUE, nugget = 20,
                 cov.model = "exponential")

kappa not used for the exponential correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.

exp.reml <- likfit(resid.geodat, ini=c(22,50/3), fix.nugget = TRUE, nugget = 20,
                 cov.model = "exponential", lik.method = "REML")

kappa not used for the exponential correlation function
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
```

```
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.

exp.ml$cov.pars

[1] 8.313124 4.934723

exp.reml$cov.pars

[1] 9.341821 5.935625

sph.ml <- likfit(resid.geodat, ini=c(22,50), fix.nugget = TRUE, nugget = 20,
                 cov.model = "sph", messages = FALSE)

sph.reml <- likfit(resid.geodat, ini=c(22,50), fix.nugget = TRUE, nugget = 20,
                 cov.model = "sph", lik.method = "REML", messages = FALSE)

sph.ml$cov.pars

[1] 27.73756 42.79786

sph.reml$cov.pars

[1] 30.04778 43.43237

gaus.ml <- likfit(resid.geodat, ini=c(22,50/sqrt(3)), fix.nugget = TRUE, nugget = 20,
                 cov.model = "gau", messages = FALSE)
gaus.reml <- likfit(resid.geodat, ini=c(22,50/sqrt(3)), fix.nugget = TRUE, nugget = 20,
                 cov.model = "gau", lik.method = "REML", messages = FALSE)

gaus.ml$cov.pars

[1] 7.738056 6.079979

gaus.reml$cov.pars

[1] 8.415170 6.315104
```



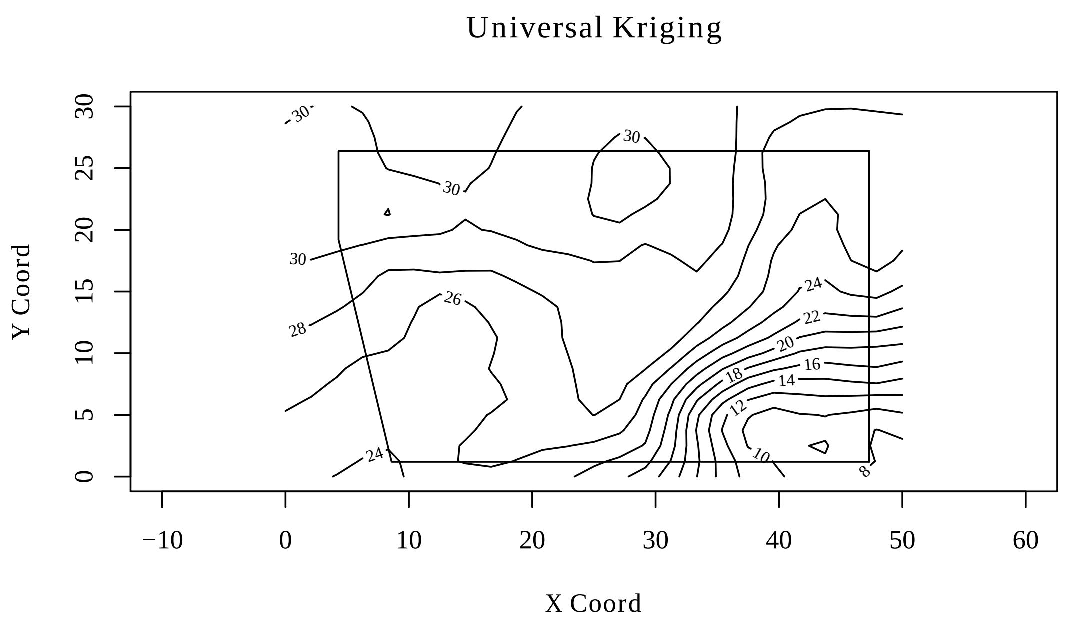
(d) *Predict yields using universal kriging and ordinary kriging. Use the parameter estimates from the residual semivariogram when you do ordinary kriging. Plot the results along with a plot of the kriging standard errors. Remember to be careful of that range parameter - what you enter depends on which semivariogram model you used. Compare the results and comment.*

Fit an OK model using the residual semivariogram parameter estimates and then fit a UK model using the same estimates.

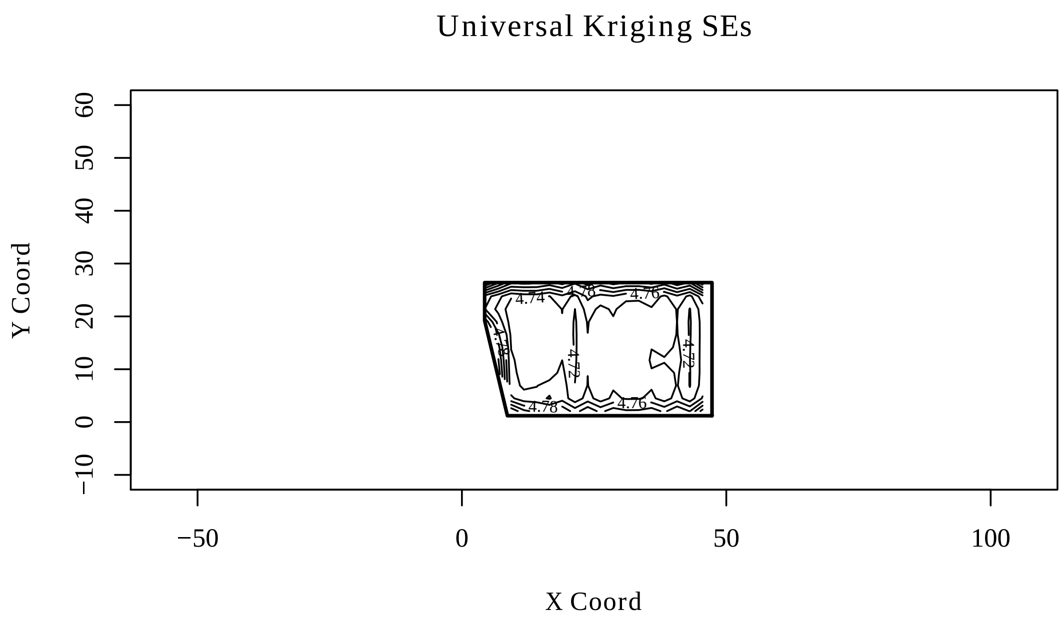
Both OK and UK results show similar rough-ness patterns in the prediction contour plot. The UK model produced more extreme predictions than the OK results. The prediction SE's are exactly the same.

```
# Note trend=2,m0="kt" in the argument list when doing universal kriging.
nugget <- sph.reml$nugget
effsill <- sph.reml$cov.pars[1] + nugget
range <- sph.reml$cov.pars[2]
wheat.uk<-ksline(wheat.geodat,locations=wheat.grid,cov.model="spherical",
cov.pars=c(effsill,range),
nugget=nugget,trend=2,m0="kt")

ksline: kriging location:  1 out of 625
ksline: kriging location:  101 out of 625
ksline: kriging location:  201 out of 625
ksline: kriging location:  301 out of 625
ksline: kriging location:  401 out of 625
ksline: kriging location:  501 out of 625
ksline: kriging location:  601 out of 625
ksline: kriging location:  625 out of 625
Kriging performed using global neighbourhood

#note hpts is not defined in the notes?
```

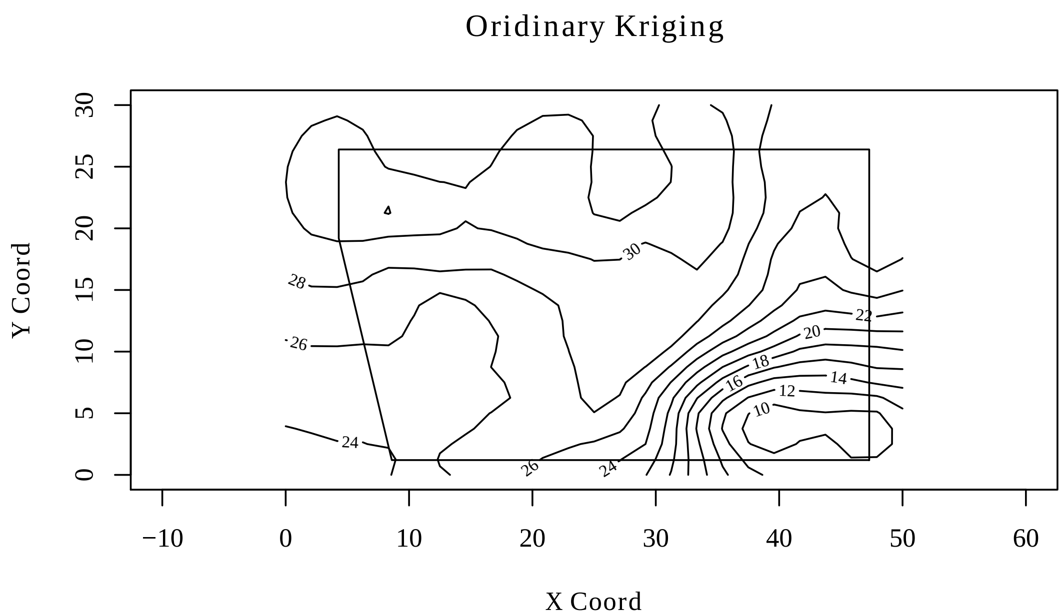
```
par(mfrow=c(1,1))
htps <- chull(wheat.dat[,1],wheat.dat[,2])
htps <- c(htps,htps[1])
contour(wheat.uk,nlevels=15, main = "Universal Kriging")
lines(wheat.dat[htps,1],wheat.dat[htps,2])
```



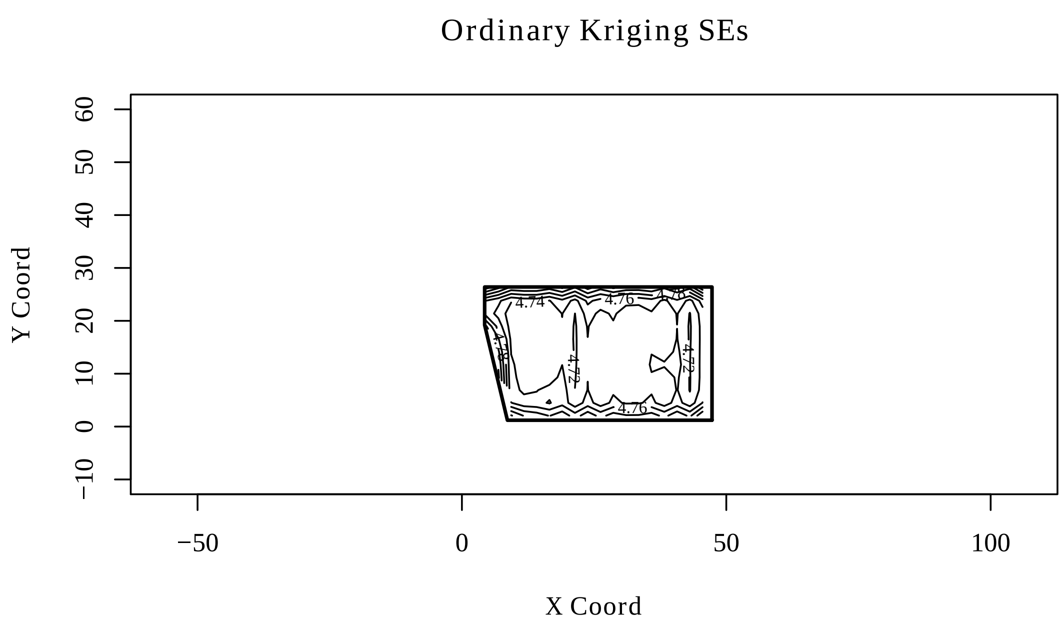

```
bord <- wheat.dat[chull(wheat.dat[,1],wheat.dat[,2]),-c(3:5)]
pred.grid<-expand.grid(seq(-10,60,l=30),seq(-10,60,l=30))

kr.border <- ksline(wheat.geodat,locations=pred.grid,
                    borders=bord,
                    cov.model = "spherical", cov.pars = c(22,50),
                    nugget = 20,m0="kt",trend=1, messages = FALSE)
contour(kr.border,val=sqrt(kr.border$krige.var))
title(main="Universal Kriging SEs")
```




```
wheat.ok<-ksline(wheat.geodat,locations=wheat.grid,cov.model="spherical",
cov.pars=c(effsill,range),nugget=nugget)

ksline: kriging location:  1 out of 625
ksline: kriging location:  101 out of 625
ksline: kriging location:  201 out of 625
ksline: kriging location:  301 out of 625
ksline: kriging location:  401 out of 625
ksline: kriging location:  501 out of 625
ksline: kriging location:  601 out of 625
ksline: kriging location:  625 out of 625
Kriging performed using global neighbourhood

contour(wheat.ok,nlevels=15, main ="Oridinary Kriging")
lines(wheat.dat[htps,1],wheat.dat[htps,2])
```




```r
kr.border.ok <- ksline(wheat.geodat,locations=pred.grid,
                      borders=bord,
                      cov.model = "spherical", cov.pars = c(22,50),
                      nugget = 20, messages = FALSE)
contour(kr.border.ok,val=sqrt(kr.border.ok$krige.var))
title(main="Ordinary Kriging SEs")
```



2.
```r
C <- function(t){
  exp(-(3*t/5))
}
```

```
dist.dat <- dist()

Error in as.matrix(x):  argument "x" is missing, with no default
```

```
u<-seq(2,4,l=1000)
m<- -(1 -t(one.vec)%*%solve(Sigmat)%*%sigvec.B)/(
  t(one.vec)%*%solve(Sigmat)%*%one.vec)

Error in t(one.vec):  object 'one.vec' not found

# approximate sigma(B,B)
# set up a storage vector which will contain the
# point to block covariances between Z((2,4)) and
# the individual points in the interval (2,4).
e<-rep(0,1000)
# Now approximate the point to block covariances
# Cov(Z(B),Z(u[i]) where u[i] is a point in B
for(i in 1:1000){
 e[i]<-mean(exp(-3*abs(u-u[i])/5))}
# now take the mean of all the point to block covariances
# in e to get Cov(Z(B),Z(B))
sig.BB<-mean(e)
sig.BB

[1] 0.6958757
```