

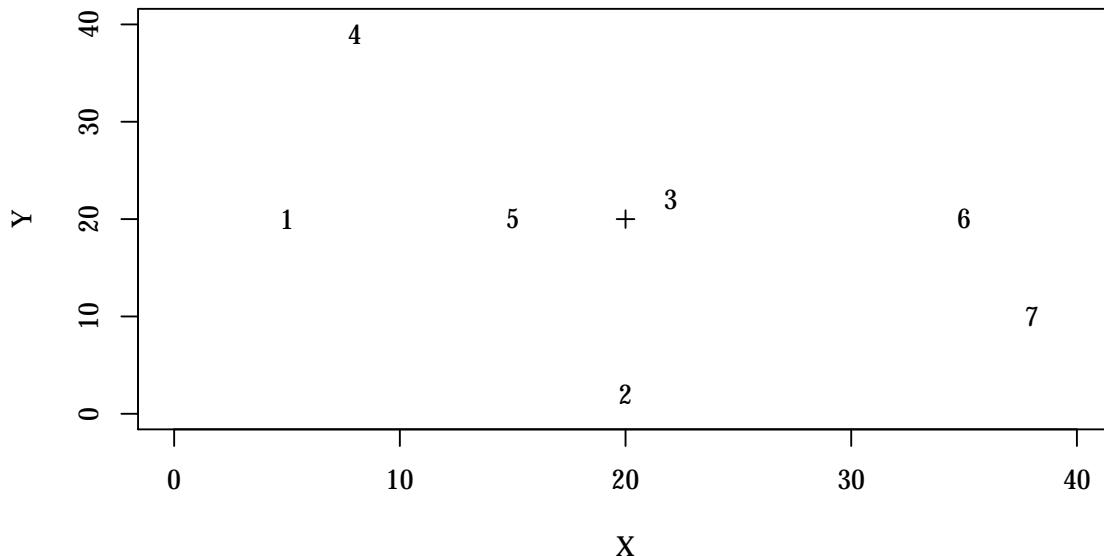
1. Refer to the 7 point ordinary kriging example we discussed in class. Repeat the example with the following changes. We move point s3 to (22,22) and point s5 to (15,20). Compare and contrast the results to the example being sure to address the following. You need to be careful with the nugget effect and range parameters.

Recall that Model D is a pure nugget effect model. You do not need the above function to fit that one. A table showing predictions, kriging variances, and weights will help.

	rho.pred	sigma2.pred	lambda1	lambda2	lambda3	lambda4	lambda5	lambda6	lambda7
A.old	66.23	9.74	0.08	0.13	0.20	0.10	0.24	0.15	0.09
B.old	69.04	11.25	0.12	0.14	0.16	0.14	0.16	0.14	0.13
C.old	66.23	4.87	0.08	0.13	0.20	0.10	0.24	0.15	0.09
D.old	70.00	10.71	0.14	0.14	0.14	0.14	0.14	0.14	0.14
E.old	66.23	19.48	0.08	0.13	0.20	0.10	0.24	0.15	0.09
F.old	44.52	6.67	-0.35	0.08	0.28	0.06	0.75	0.18	0.01
			15.00	18.00	13.00	22.47	10.44	15.00	20.59

	rho.pred	sigma2.pred	lambda1	lambda2	lambda3	lambda4	lambda5	lambda6	lambda7
A.new	58.83	5.02	0.02	0.04	0.57	0.02	0.30	0.03	0.03
B.new	62.59	8.11	0.06	0.06	0.46	0.06	0.23	0.06	0.06
C.new	58.83	2.51	0.02	0.04	0.57	0.02	0.30	0.03	0.03
D.new	70.00	10.71	0.14	0.14	0.14	0.14	0.14	0.14	0.14
E.new	58.83	10.04	0.02	0.04	0.57	0.02	0.30	0.03	0.03
F.new	51.45	0.36	-0.09	0.02	0.68	-0.01	0.41	-0.03	0.01
			15.00	18.00	2.83	22.47	5.00	15.00	20.59

Locations



- (a) *Do the same conclusions regarding the effects of the sill, range, and nugget effect still hold?*

Yes, the effects still hold.

Effect of Sill

The relative sill, range, and nugget parameters are the same with the model with the two locations changed as with the original model.

Model A and Model E both have a practical range of 20 and nugget of 0, with an exponential covariance model. Model A has half the sill of model E. Yes, the effects on predicts and kriging variance stay the same as before. The predictions ad weights are the same with Models A and E and the kriging variance of Model E is twice the kriging variance of Model A, as with the original locations.

Effect of (Practical) Range

Models A and B both have nugget effects of 0 and sills of 10 with an exponential specified covariance model. The (practical) range is 20 for Model A and 10 for Model B. The predictions, kriging variance, and weights are all different between the two models, but again have similar effects as those in the original model. Model B assigns more weight to locations farther away from the prediction location and less weight to locations closer to the prediction location than Model A does. This makes sense because specifying a larger practical range takes more information from points farther away. Likewise, again we see the kriging variance increase for Model B where there is a larger practical range.

Effect of Nugget

Models A and C both have ranges of 20 and sills of 10. The nugget for Model A is 0 while the nugget for Model C is 5. Model D is a pure nugget model. The weights and predictions the same between the two models. The variance without a nugget effect is twice the variance with the nugget effect because with a sill of 10 in both and a nugget of 5, half the total variation is absorbed by the nugget, removing half the variation that could be attributed to spatial correlations. These conclusions are again the same as with the original data. We see the pure nugget model (Model D) differs from both Model A and Model B in predictions, kriging variance, and weights. With a pure nugget effect, all locations have equal weights and the variance is much larger.

- (b) *Which points are screening?*

Points 3 and 5 are screening as they have larger weights and are at closer distances.

- (c) *What happens to the kriging variance?*

The kriging variance is smallest with the gaussian model, although the gaussian model does return negative weights.

The kriging variance decreases in the presence of a nugget, increases with a decrease in practical range (less information), increases with an increased sill, and in the absence of spatial correlation

is very close to the nugget.

- (d) *In which models does the predicted value change and in which does it stay the same?*

Models A, C and E all predicted the same value. These models all had the same practical range and were fit with the exponential model.

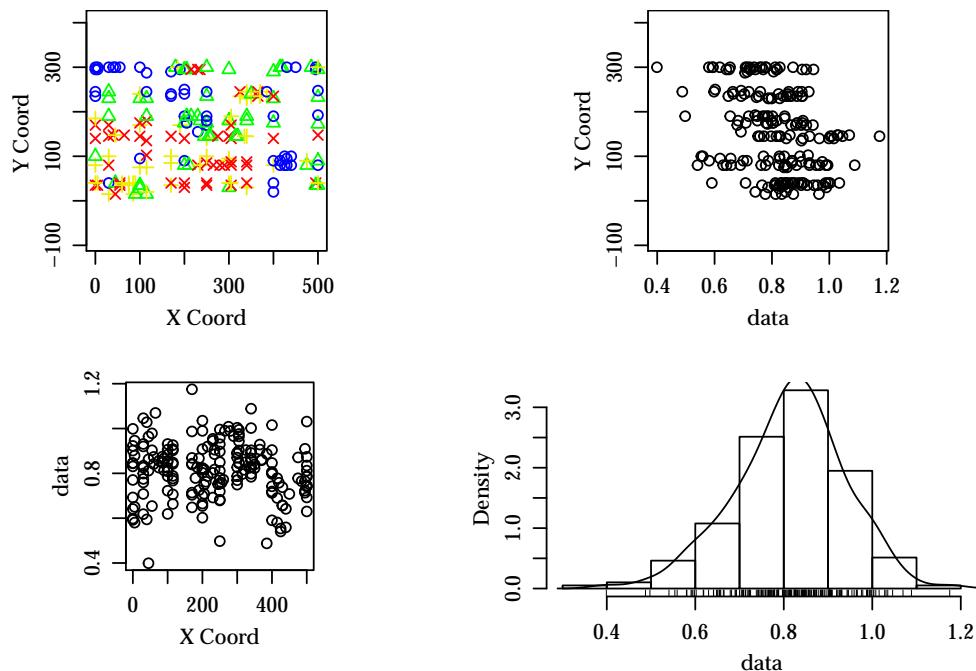
Models B, D, and F all had different predicted values. Models D and F were fit with different models (nugget and gaussian, respectfully) and Model B had half the practical range as models A, C, and E.

2. *We are going to use ordinary kriging to predict values of total carbon over a grid. You should still have the carbon/nitrogen data set from the last homework but let me know if you need to have it emailed to you. Choose your grid to be the same as in the CN ratio example I worked in class. Set up R code is below.*

We will work with total carbon (column labelled TC in the data set) in this problem. You can use either the ksline or krige.conv functions. Be a bit patient; it should take only a minute or so. Dont worry about anisotropy.

- (a) *Plot the data. Do you see any evidence of trend? Outliers? Other potential anomalies?*

The data appear to be gridded. Trends do not appear in either the north-south nor the east-west directions. Trying very hard to make a trend, you might say there's a very slight negative linear trend in the north-south direction. A few outliers stand out appear in both vertical and horizontal directions. The shapes (colors) which represent the quartiles in the top left plot appear to mostly occur in clusters. This means that data in similar quantiles appear to be located closer together. The data are normally distributed, with a very minimal left skew.

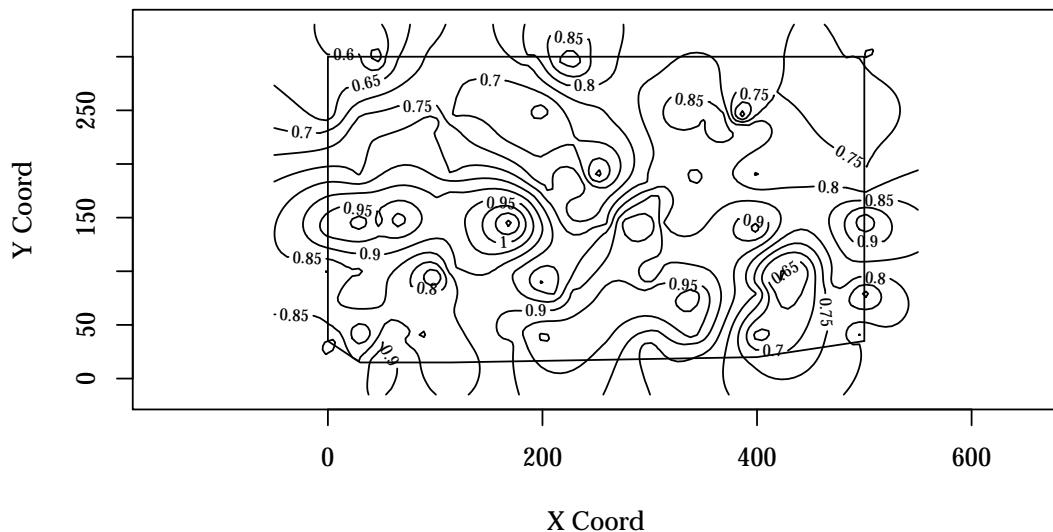


- (b) Predict using global ordinary kriging. You estimated a semivariogram for these data on the last homework and you can use those results. Prepare contour plots of the predictions and kriging standard errors.

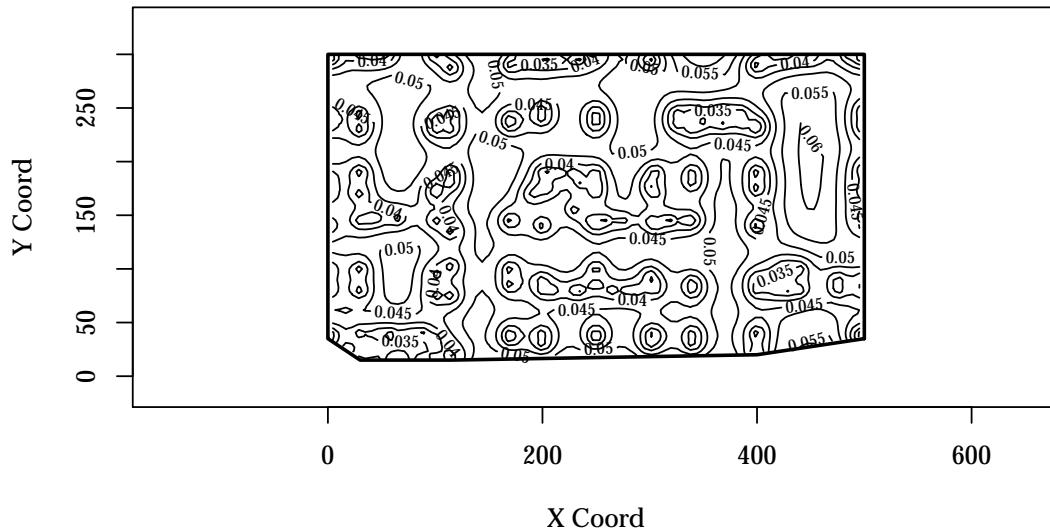
```

variog: computing omnidirectional variogram
variofit: covariance model used is exponential
variofit: weights used: equal
variofit: minimisation function used: optim
  
```

Predictions



Global Kriging SEs



- (c) Evaluate the predictions using cross validation (CV2). You can use the R code I provided in class notes.

I also computed CV1, which is quite closer to 0, than CV2 is to 1.

Having CV2 close to 1 means that the \sqrt{MSEP} is unbiased for the kriging variance among all j location removals. CV2 is above 1, meaning that the \sqrt{MSEP} is a biased measure for the kriging variance. The predictions are off more than expected.

CV1	CV2
0.01	1.53

- (d) Briefly discuss your results.

The prediction plot shows quite a few contours with step slopes, and so is not that smooth. The global kriging SE's were all not very large, but also have quite a few steep contours. Prediction at locations with steep contours will change rapidly even when the SE's are not very large, but may explain some of the bias suggested by the CV2 value.

We do see the lowest predictions are in the upper left and lower right corners, as was suggested by the colored plot of the original data where the clusters with the lowest quantile are at.

In part

R Code

Problem 1

```

x <- c(5,20,25,8,10,35,38)
y <- c(20,2,32,39,17,20,10)
z <- c(100,70,60,90,50,80,40)

loc <- data.frame(cbind(x,y,z))
colnames(loc) <- c("x", "y", "z")

loc.new <- loc
loc.new[3,c(1,2)] <- c(22,22)
loc.new[5,c(1,2)] <- c(15,20)

dist.new <- as.matrix(dist(loc.new[,1:2]))

sigma.vec.new <- as.matrix(dist(rbind(loc.new[,1:2],c(20,20))))
sigma.vec.new <- sigma.vec.new[8,]
sigma.vec.new[8] <- 1

sigma.vec.old <- as.matrix(dist(rbind(loc[,1:2],c(20,20))))
sigma.vec.old <- sigma.vec.old[8,]
sigma.vec.old[8] <- 1

x <- c(5,20,25,8,10,35,38)
y <- c(20,2,32,39,17,20,10)
z <- c(100,70,60,90,50,80,40)

loc <- data.frame(cbind(x,y,z))
colnames(loc) <- c("x", "y", "z")

loc.new <- loc
loc.new[3,c(1,2)] <- c(22,22)
loc.new[5,c(1,2)] <- c(15,20)

dist.new <- as.matrix(dist(loc.new[,1:2]))

sigma.vec.new <- as.matrix(dist(rbind(loc.new[,1:2],c(20,20))))
sigma.vec.new <- sigma.vec.new[8,]
sigma.vec.new[8] <- 1

sigma.vec.old <- as.matrix(dist(rbind(loc[,1:2],c(20,20))))
sigma.vec.old <- sigma.vec.old[8,]
sigma.vec.old[8] <- 1

dist.old <- as.matrix(dist(loc[,1:2]))
sigma.vec <- as.matrix(dist(rbind(loc[,1:2],c(20,20))))
sigma.vec <- sigma.vec[8,]
sigma.vec[8] <- 1

# try to automate
dmat.old <- as.matrix(dist.old)
dmat.new <- as.matrix(dist.new)

sig.fn <- function(dmat, model, sill, nugget, range, offset, sigma.vec, location){
  Sigma <- cov.spatial(dmat, cov.model = model,
                        cov.pars=c(sill - nugget, range/offset))
  pred <- c(rep(1,7))
  Sigma.star <- data.frame(cbind(Sigma,pred))
}

```

```

pred <- c(rep(1,7),0)
Sigma.star[8,] <- c(pred)

# find sigma star
sigma.star <- cov.spatial(sigma.vec,cov.model=model, cov.pars=c((sill - nugget), range/offset))
sigma.star[8] <- 1
sigma.star <- t(as.matrix(sigma.star))

lambda.star <- sigma.star %*% solve(Sigma.star)

pred.new <- sum(lambda.star[-8]*location[,3])
var.new <- sill - nugget - sum(lambda.star*sigma.star)

return(c(pred.new, var.new, c(lambda.star)[-8]))
}

# dmat, model, sill, nugget, range, offset, sigma.vec, location

A.old <- sig.fn(dmat = dist.old, sill = 10,
                  nugget = 0, range = 20, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.old,
                  location = loc)

B.old <- sig.fn(dmat = dist.old, sill = 10,
                  nugget = 0, range = 10, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.old,
                  location = loc)

C.old <- sig.fn(dmat = dist.old, sill = 10,
                  nugget = 5, range = 20, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.old,
                  location = loc)

Sigma.oldD <- diag(5,7,7)
pred <- c(rep(1,7))
Sigma.star.oldD <- data.frame(cbind(Sigma.oldD,pred))
pred <- c(rep(1,7),0)
Sigma.star.oldD[8,] <- c(pred)

# find sigma star
sigma.star.oldD <- c(rep(0,7),1)
sigma.star.oldD <- t(as.matrix(sigma.star.oldD))

lambda.star.oldD <- sigma.star.oldD %*% solve(Sigma.star.oldD)

pred.oldD <- sum(lambda.star.oldD[-8]*loc[,3])
var.oldD <- 10-sum(lambda.star.oldD*sigma.star.oldD)

D.old <- c(pred.oldD, var.oldD,
            c(lambda.star.oldD[-8]))

E.old <- sig.fn(dmat = dist.old, sill = 20,
                  nugget = 0, range = 20, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.old,
                  location = loc)

F.old <- sig.fn(dmat = dist.old, sill = 10,

```

```

nugget = 0, range = 20, offset = sqrt(3),
model = "gaussian",
sigma.vec = sigma.vec.old,
location = loc)

dist.old.out <- as.matrix(dist(data.frame(rbind(loc[,1:2],c(20,20)))))[8,-8]

old <- data.frame(rbind(A.old, B.old, C.old, D.old, E.old,
                         F.old, c("", "", dist.old.out)))
colnames(old) <- c("rho.pred", "sigma2.pred", "lambda1", "lambda2", "lambda3",
                   "lambda4", "lambda5", "lambda6", "lambda7")

a <- rownames(old)

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

rownames(old) <- c(a)

print(xtable(old, align = "| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |"))

A.new <- sig.fn(dmat = dist.new, sill = 10,
                  nugget = 0, range = 20, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.new,
                  location = loc.new)

B.new <- sig.fn(dmat = dist.new, sill = 10,
                  nugget = 0, range = 10, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.new,
                  location = loc.new)

C.new <- sig.fn(dmat = dist.new, sill = 10,
                  nugget = 5, range = 20, offset = 3,
                  model = "exponential",
                  sigma.vec = sigma.vec.new,
                  location = loc.new)

Sigma.newD <- diag(5,7,7)
pred <- c(rep(1,7))
Sigma.star.newD <- data.frame(cbind(Sigma.newD,pred))
pred <- c(rep(1,7),0)
Sigma.star.newD[8,] <- c(pred)

# find sigma star
sigma.star.newD <- c(rep(0,7),1)
sigma.star.newD <- t(as.matrix(sigma.star.newD))

lambda.star.newD <- sigma.star.newD %*% solve(Sigma.star.newD)

pred.newD <- sum(lambda.star.newD[-8]*loc.new[,3])
var.newD <- 10-sum(lambda.star.newD*sigma.star.newD)

D.new <- c(pred.newD, var.newD,
            c(lambda.star.newD[-8]))

E.new <- sig.fn(dmat = dist.new, sill = 20,
                  nugget = 0, range = 20, offset = 3,

```

```

model = "exponential",
sigma.vec = sigma.vec.new,
location = loc.new)

F.new <- sig.fn(dmat = dist.new, sill = 10,
nugget = 0, range = 20, offset = sqrt(3),
model = "gaussian",
sigma.vec = sigma.vec.new,
location = loc.new)

dist.new.out <- as.matrix(dist(data.frame(rbind(loc.new[,1:2],c(20,20)))))[,8,-8]

new.out <- data.frame(rbind(A.new, B.new, C.new, D.new, E.new,F.new,c("", "", dist.new.out)))
colnames(new.out) <- c("rho.pred", "sigma2.pred", "lambda1", "lambda2", "lambda3",
"lambda4", "lambda5", "lambda6", "lambda7")

b <- rownames(new.out)

new.out <- data.frame(apply(new.out, 2, function(t){
  round(as.numeric(t),2)
}))

rownames(new.out) <- c(b)

print(xtable(new.out, align = "| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |"))

names <- c("1", "2", "3", "4", "5", "6", "7")
plot(x=loc.new$x,y=loc.new$y, pch=names, xlim = c(0,40), ylim=c(0,40), main = "Locations",
xlab = "X", ylab = "Y")
points(20,20,pch=3)

```

Problem 2

```

CN.dat <- read.table("CN.dat", header = TRUE)

pred.grid<-expand.grid(seq(-50,550,l=100),seq(-15,330,l=100))

TC.geodat<-as.geodata(CN.dat,coords.col=1:2,data.col=4)

plot(TC.geodat)

TC.var <- variog(coords = TC.geodat$coords, data = as.numeric(TC.geodat$data))
ols.n <- variofit(TC.var, ini = c(0.011,150/3), weights = "equal",
nugget=0.005, cov.model = "exponential")

tc.global <- ksline(TC.geodat,locations = pred.grid, cov.model = "exponential", m0="kt",
cov.pars = c(0.01116,150),nugget = 0.0005, trend = 1, messages = FALSE)

contour(tc.global) #levels
title(main = "Predictions")

https <- chull(CN.dat[,1],CN.dat[,2])
https <- c(https,https[1])
lines(CN.dat[https,1],CN.dat[https,2])

bord <- CN.dat[chull(CN.dat[,1],CN.dat[,2]),-c(3:5)]
kr.border <- ksline(TC.geodat,locations=pred.grid,
borders=bord,
cov.model = "exponential", cov.pars = c(0.01116,150),
nugget = 0.0005,m0="kt",trend=1, messages = FALSE)
contour(kr.border,val=sqrt(kr.border$krige.var))

```

```
title(main="Global Kriging SEs")

p.xval <- numeric()
s.xval <- numeric()

for(i in 1:dim(TC.geodat$coord)[1]){
  a <- ksline(coords=CN.dat[-i,1:2], data = CN.dat[-i,4],
               locations=CN.dat[i,1:2], cov.pars = c(0.01116,150/3),
               nugget = 0.0005, cov.model = "exponential",
               m0="kt", trend = 1, messages = FALSE)
  p.xval[i] <- a$predict
  s.xval[i] <- sqrt(a$krige.var)
}

b <- (CN.dat[,4] - p.xval)/s.xval
cv1 <- mean(b)
cv2 <- sqrt(mean(b^2))
```