Chapter Five Notes

We have looked at derivation of the kriging equations for simple and ordinary kriging. We will look at a simple example.

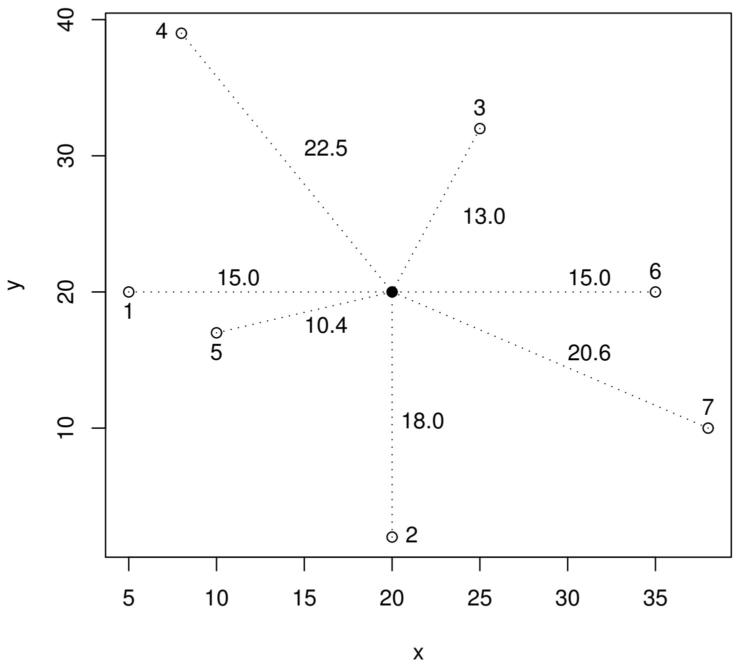
Example 5.5 in Chapter 5

We have 7 locations with the following Z values.

i	$\mathbf{s}_i$	Z
1	$(5, 20)$	100
2	$(20, 2)$	70
3	$(25, 32)$	60
4	$(8, 39)$	90
5	$(10, 17)$	50
6	$(35, 20)$	80
7	$(38, 10)$	40

The location at which we wish to predict has coordinates $\mathbf{s}_0 = (20, 20)$. The plot below is a reproduction of Figure 5.2 from the text showing the locations of the points and the distance each lies from the location at which we wish to predict.

Figure 5.2 on page 230



They predict $Z(\mathbf{s}_0)$ using 6 different covariance models summarized in the table below.

Model	Range	Sill	Nugget	Type
A	20	10	0	Exponential
B	10	10	0	Exponential
C	20	10	5	Exponential
D	–	–	10	Nugget
E	20	20	0	Exponential
F	20	10	0	Gaussian

The authors summarize the results in Table 5.1 on page 231. I will reproduce the results for the first covariance model using R. The first is an exponential covariance model with sill of 10 and (practical) range of 20. The covariance function is

$$C(h) = 10\left[1 - \exp\left(-3h/20\right)\right].$$

Recall that the ordinary kriging predictor

$$p_{ok}\left(\mathbf{Z}; \mathbf{s}_0\right) = \widehat{\mu} + \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\left(\mathbf{Z}\left(\mathbf{s}\right) - \mathbf{1}\widehat{\mu}\right)$$

where

$$\widehat{\mu} = \left(\mathbf{1}'\ \boldsymbol{\Sigma}^{-1}\mathbf{1}\right)^{-1}\mathbf{1}'\ \boldsymbol{\Sigma}^{-1}\mathbf{Z}\left(\mathbf{s}\right).$$

Actually, it is easier to get the predicted value than that:

$$p_{ok}\left(\mathbf{Z}; \mathbf{s}_0\right) = \sum_{i=1}^{n} \lambda_i Z\left(\mathbf{s}_i\right).$$

```
# create distance matrix
dmat<-as.matrix(dist(examp5.dat[,1:2]))
dmat
```

```
 0.000000   23.43075   23.32381   19.23538    5.830952   30.00000   34.48188
23.430749    0.00000   30.41381   38.89730   18.027756   23.43075   19.69772
23.323808   30.41381    0.00000   18.38478   21.213203   15.62050   25.55386
19.235384   38.89730   18.38478    0.00000   22.090722   33.01515   41.72529
 5.830952   18.02776   21.21320   22.09072    0.000000   25.17936   28.86174
30.000000   23.43075   15.62050   33.01515   25.179357    0.00000   10.44031
34.481879   19.69772   25.55386   41.72529   28.861739   10.44031    0.00000
```

This matrix contains the Euclidean distances between the 7 points at which observations were made. For example, the first point with coordinate $(5, 20)$ and the second point with coordinates $(20, 2)$ lie

$$h_{12}\sqrt{(5-20)^2 + (20-2)^2} = 23.431$$

units apart.

We can easily create the covariance matrix $\boldsymbol{\Sigma}$.

```
Sigma<-cov.spatial(dmat,cov.model="exponential",cov.pars=c(10,20/3))
Sigma
```

```
10.00000000  0.29759337   0.3024056  0.55837611   4.1701096   0.11108997   0.05671612
 0.29759337 10.00000000   0.1044041  0.02924607   0.6692629   0.29759337   0.52096509
 0.30240562  0.10440405  10.0000000  0.63436464   0.4150338   0.96031895   0.21642859
 0.55837611  0.02924607   0.6343646 10.00000000   0.3638465   0.07067332   0.01913553
 4.17010956  0.66926287   0.4150338  0.36384650  10.0000000   0.22893472   0.13177283
 0.11108997  0.29759337   0.9603190  0.07067332   0.2289347  10.00000000   2.08869426
 0.05671612  0.52096509   0.2164286  0.01913553   0.1317728   2.08869426  10.00000000
```

For example, we have

$$C\left(h_{12}\right) = C(23.471) = 10\left[1 - \exp\left(-3(23.471)/20\right)\right] = 0.2976.$$

Recall that the solution in matrix format is

$$\boldsymbol{\lambda}^{*} = \boldsymbol{\Sigma}^{*-1}\boldsymbol{\sigma}^{*}$$

where

$$\boldsymbol{\Sigma}^{**} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} & 1 \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} = \begin{bmatrix} 10.000 & 0.298 & 0.302 & 0.558 & 4.170 & 0.111 & 0.057 & 1 \\ 0.298 & 10.000 & 0.104 & 0.029 & 0.669 & 0.298 & 0.521 & 1 \\ 0.302 & 0.104 & 10.000 & 0.634 & 0.415 & 0.960 & 0.216 & 1 \\ 0.558 & 0.029 & 0.634 & 10.000 & 0.364 & 0.071 & 0.019 & 1 \\ 4.170 & 0.669 & 0.415 & 0.364 & 10.000 & 0.229 & 0.132 & 1 \\ 0.111 & 0.298 & 0.960 & 0.071 & 0.229 & 10.000 & 2.089 & 1 \\ 0.057 & 0.521 & 0.216 & 0.019 & 0.132 & 2.089 & 10.000 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{\lambda}^{*} = \left[\lambda_1 \;\; \lambda_2 \quad \cdots \;\; \lambda_n \;\; m\right]'$$

and

$$\boldsymbol{\sigma}^{*} = \begin{bmatrix} \sigma_{10} \\ \sigma_{20} \\ \vdots \\ \sigma_{n0} \\ 1 \end{bmatrix} = \begin{bmatrix} 1.054 \\ 0.672 \\ 1.423 \\ 0.344 \\ 2.089 \\ 1.054 \\ 0.456 \\ 1 \end{bmatrix}$$

The covariances in $\boldsymbol{\sigma}^{*}$ were calculated with

```
sigma.vec<-as.matrix(dist(rbind(examp5.dat[,1:2],c(20,20))))
sigma.vec<-sigma.vec[8,]
sigma.star<-cov.spatial(sigma.vec,cov.model="exponential",cov.pars=c(10,20/3))
sigma.star[8]=1}
```

The solution was then calculated.

```
lambda.star<-solve(Sigma.star)%*%sigma.star
lambda.star
1 0.0800
2 0.1311
3 0.1999
4 0.1011
5 0.2444
6 0.1499
7 0.0936
  -0.9436
```

The weights are the first 7 elements and the eighth element is the Lagrange multiplier quantity m. The predicted value is

$$p_{ok}\left(\mathbf{Z};\mathbf{s}_0\right)=\sum_{i=1}^{n}\lambda_i Z\left(\mathbf{s}_i\right)=(0.08)(100)+\cdots+(0.09)(40)=66.23.$$

The kriging variance is

$$\begin{aligned}\sigma^2_{ok} &= C(\mathbf{0})-\boldsymbol{\lambda}'\boldsymbol{\sigma}-m\\ &= C(\mathbf{0})-\sum_{i=1}^{n}\lambda_i\sigma_{i0}-m\\ &= 10+(0.08)(1.054)+\cdots+(0.09)(0.456)-(-0.9436)\\ &= 9.74\end{aligned}$$

```
sum(lambda.star[-8]*examp5.dat[,3])
 66.22654
10-sum(lambda.star*sigma.star)
 9.740824
```

The table below is Table 5.1 in the text.

Model	p_{ok}	σ^2_{ok}	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
A	66.23	9.74	0.08	0.13	0.20	0.10	0.24	0.15	0.09
B	69.04	11.25	0.12	0.14	0.16	0.14	0.16	0.14	0.13
C	68.64	10.63	0.12	0.14	0.17	0.12	0.18	0.14	0.12
D	70.00	11.43	0.14	0.14	0.14	0.14	0.14	0.14	0.14
E	66.23	19.48	0.08	0.13	0.20	0.10	0.24	0.15	0.09
F	44.52	6.67	−0.35	0.08	0.28	0.06	0.75	0.18	0.01
	$\Vert\mathbf{s}_0-\mathbf{s}_i\Vert$		15.0	18.0	13.0	22.5	10.4	15.0	20.6

Different covariance (semivariogram) models produce different weights, predicted values, and kriging variances. We would like to understand a bit more about why that happens.

Consider Model A. The weights are found as a solution to the equation

$$\boldsymbol{\lambda}^* = \boldsymbol{\Sigma}^{*-1}\boldsymbol{\sigma}^*$$

where

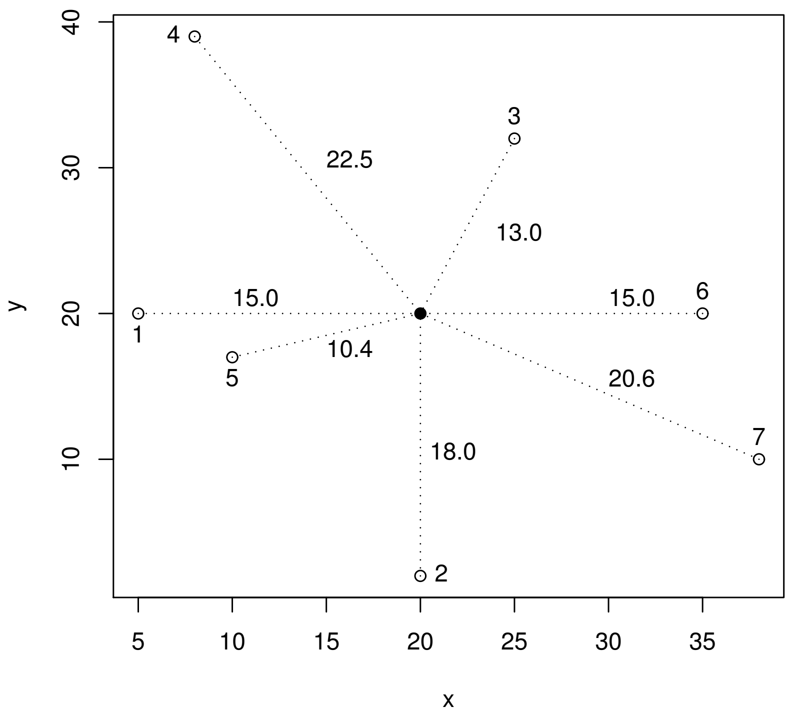
$$\boldsymbol{\Sigma}^* = \begin{bmatrix} 10.000 & 0.298 & 0.302 & 0.558 & 4.170 & 0.111 & 0.057 & 1 \\ 0.298 & 10.000 & 0.104 & 0.029 & 0.669 & 0.298 & 0.521 & 1 \\ 0.302 & 0.104 & 10.000 & 0.634 & 0.415 & 0.960 & 0.216 & 1 \\ 0.558 & 0.029 & 0.634 & 10.000 & 0.364 & 0.071 & 0.019 & 1 \\ 4.170 & 0.669 & 0.415 & 0.364 & 10.000 & 0.229 & 0.132 & 1 \\ 0.111 & 0.298 & 0.960 & 0.071 & 0.229 & 10.000 & 2.089 & 1 \\ 0.057 & 0.521 & 0.216 & 0.019 & 0.132 & 2.089 & 10.000 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

and

$$\boldsymbol{\sigma}^* = [1.054 \;\; 0.672 \;\; 1.423 \;\; 0.344 \;\; 2.089 \;\; 1.054 \;\; 0.456 \;\; 1]'$$

These matrices are distance matrices with distances as covariances. Points that are closer together have Z values with larger covariances. For example, points 1 and 7 lie far apart and $\sigma_{17} = 0.057$ while points 1 and 5 are close together with $\sigma_{15} = 4.170$. Similarly points 0 and 4 are far apart with $\sigma_{40} = 0.344$ while points 0 and 5 are close together with $\sigma_{50} = 2.089$. It might appear that we could just use the "distances" in $\boldsymbol{\sigma^*}$ as our weights. What do we gain by multiplying this by $\boldsymbol{\Sigma}^{*-1}$? We take better advantage of the information in the sample values in that we consider not only the Z values but the spatial arrangement of those values. First, lets take another look at the figure above.

Figure 5.2 on page 230



Note that points 1 and 6 lie an equal distance from point 0 both in terms of Euclidean distance (15) and in terms of the covariances ($\sigma_{10} = \sigma_{60} = 1.054$). But these covariances do not take into account locations near points 1 and 6. For example points 1 and 5 are close together (5.83) with 5 closer to 0 than 1. The closest point to 6 is point 7 (10.44) and there are no points in the immediate neighborhood closer to 0 than 6. Intuitively (or maybe not), point 6 has more information about $Z(\mathbf{s}_0)$ than point 1. Imagine that someone told you that you had to throw out either point 1 or point 6. Which would you choose to delete? If we throw out point 1 we still have point 5 nearby to pick up some of the slack, whereas we do not have as much backup information if we lose point 6. We need some way to take advantage of this type of information and $\mathbf{\Sigma}^{*-1}$ does just that providing information on clustering among the sample values. Note that multiplying $\boldsymbol{\sigma}^*$ by $\mathbf{\Sigma}^{*-1}$ reallocates some of the weight assigned to point 1 by σ_{10} to other points that may be as close to, or even further away from, point 0 but which are less redundant. For example $\lambda_6 = 0.15$ and $\lambda_4 = 0.10$ whereas $\lambda_1 = 0.08$ for Model A. The effect of point 1 is downweighted because of the effect of point 5. We say that the effect of point 1 is *screened* by the effect of point 5.

Effect of Sill, Range, Nugget, and Model

- *Effect of Sill (Scale)*: Consider Models A and E. Model A is an exponential model with practical range of 20 and sill of 10. Model A is an exponential model with practical range of 20 and sill of 20. Notice that the weights (and hence the predicted value) are the same but the kriging variance of Model E is twice the kriging variance of Model A. This makes sense. The more variable a random field the more uncertainty about our predictions.

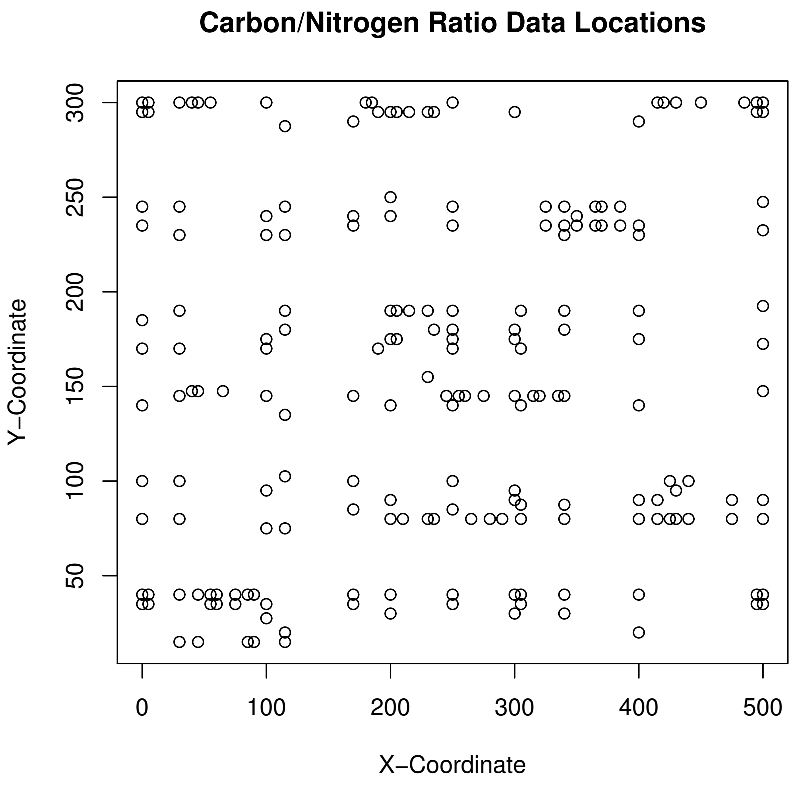
- *Effect of Model (Shape)*: Consider Models A and F. The nugget effect, practical range, and sill are identical in these models but A is an exponential model whereas F is a gaussian model. The gaussian model assumes a very continuous Z surface and assigns higher weights to closer observations. Note, for example that $\lambda_5 = 0.75$ for the gaussian model but $\lambda_5 = 0.24$ for the exponential model. Note that $\lambda_1 = -0.35$ for the gaussian model, an extreme example of screening. This is not as bad as it might seem. Negative weights and weights greater than 1 allow for the possibility of predictions smaller than the minimum and larger than the maximum values of Z observed. Negative weights are more likely with models that assume smoother Z surfaces, however. Note the differences in predicted values and kriging variances, also.

- *Effect of the Nugget*: Consider Models A, C, and D. Models A and C are both exponential models with sills of 10 and practical ranges of 20, but Model C has a nugget effect of 5 whereas Model A has no nugget effect. The weights associated with Model C are less variable than the weights assigned by Model A. A nugget effect implies less correlation at shorter distances. In effect, the larger the nugget effect the closer ordinary kriging prediction comes to simple averaging (which is exactly what we see with the pure nugget effect in Model D). Note that the predicted value in the pure nugget model is just the sample average of the 7 observations. A pure nugget effect implies that the data are uncorrelated. The BLUP is the sample mean and the prediction variance is $C(\mathbf{0})\,(1 + 1/n)$ which is $10(8/7) = 11.43$ for our example.

- *Effect of the (Practical) Range*: Consider Models A and B. They are both exponential models with nugget effects of 0 and sills of 10. The practical range of Model A is 20 and it is 10 for Model B. Thus, Model B assumes that the correlation dies off more quickly than Model A does. Model A assigns higher weights to closer observations than Model B (e.g. $\lambda_5 = 0.24$ for Model A but $\lambda_5 = 0.16$ for Model B) and less weight to more distant observations than Model B (e.g. $\lambda_4 = 0.10$ for Model A but $\lambda_4 = 0.14$ for Model B). We see that predictions change accordingly along with the kriging variance. It is higher for the Model B prediction. This makes sense. Stronger correlations should lead to more certainty in predictions.

- *CN Ratio Example - Ordinary Kriging*

  The data were described in Example on page 156. The data are 195 observed values of Carbon/Nitrogen ratios collected in an agricultural field. The locations are plotted below.



We fit several semivariogram models to the data set and concluded that an exponential model with a nugget effect fit "best" although the effective range varied quite a bit depending on the method of fitting. We will start with an isotropic exponential model with a nugget effect of 0.1132, a partial sill of 0.2023, and a range of 141.03.

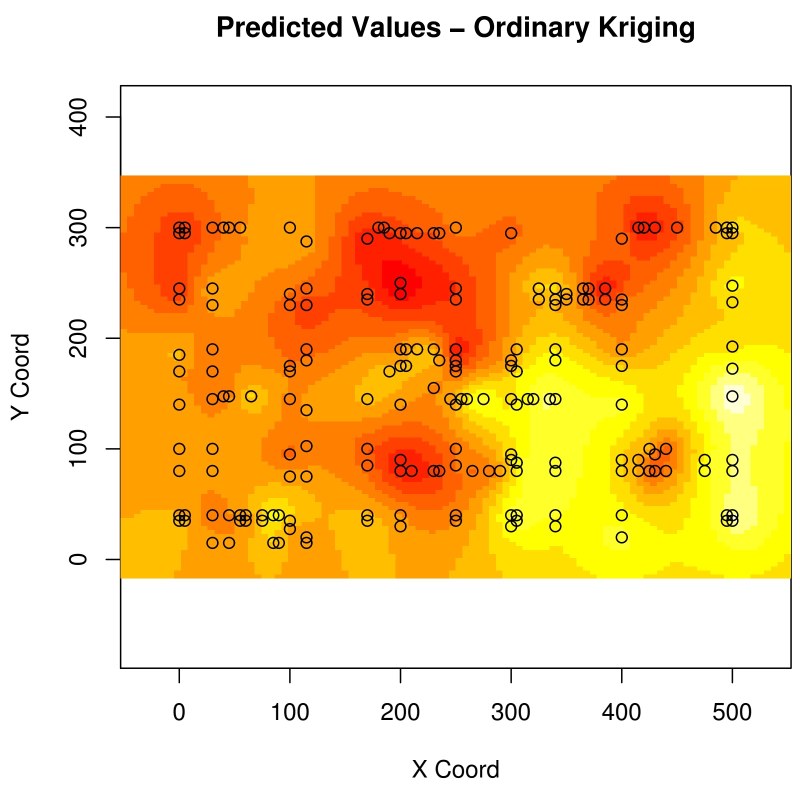
The first requirement is specification of locations at which we wish to predict CN ratios. We could predict at the 195 locations at which data were collected but remember that ordinary kriging "honors the data", i.e. it is an exact interpolator. We specify a regular grid of points at which we want to predict below.

Kriging is carried out in `geoR` using the `krige.conv` function. This function kriges globally in that all points are used. It can be more efficient to krige locally by using only points in a specified neighborhood of the prediction locations. The `krige.var` function appears to be fairly fast however allowing prediction at a large number of points relatively quickly.

```
# set up a grid
pred.grid<-expand.grid(seq(-50,550,l=100),seq(-15,330,l=100))
# ordinary kriging
CN.krige<-krige.conv(CN.geodata,locations=pred.grid,
 krige=krige.control(cov.model="exponential",
 cov.pars=c(0.215,171.1/3),nugget=0.118))
```

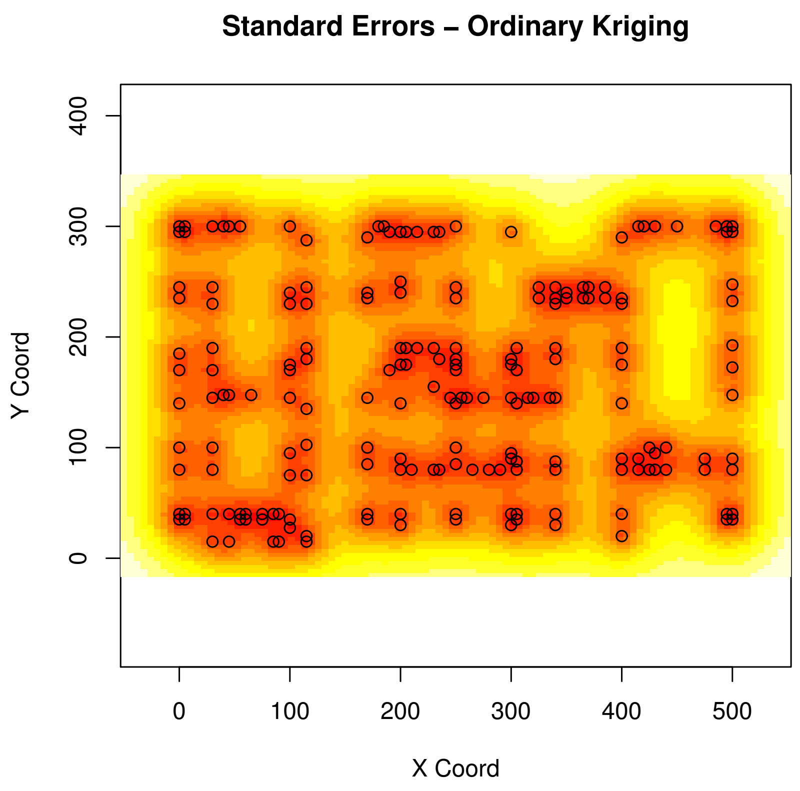
The grid has $10,000$ points but `krige.conv` only took about 20 seconds or so to run. Note that I expanded the prediction region slightly beyond the boundary in which the data occurred. The results are generally viewed graphically.

```
image(CN.krige)
points(CN.dat[,1],CN.dat[,2])
title(main="Predicted Values - Ordinary Kriging")
```



The points at which observations were made are overlaid on the image. It is always worth looking at the kriging variances or standard errors.

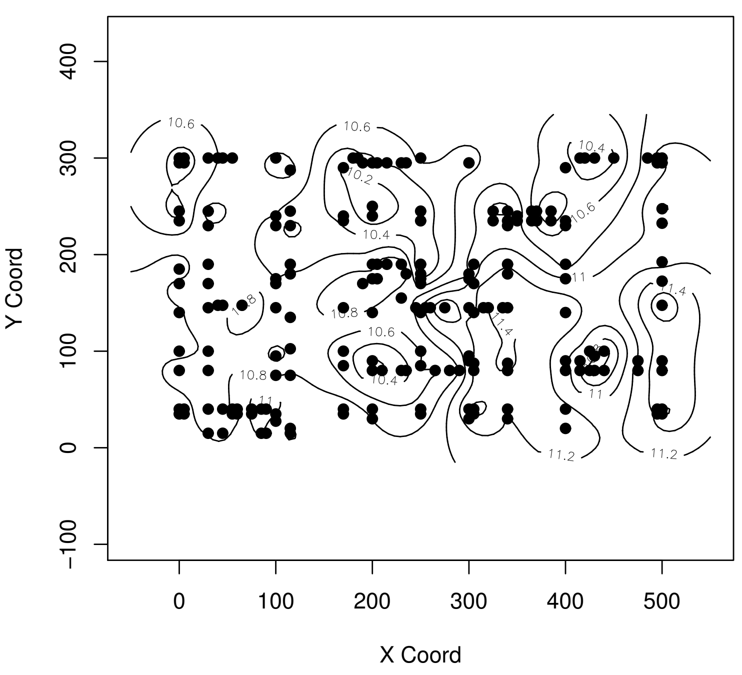
```
image(CN.krige,val=sqrt(CN.krige$krige.var))
points(CN.dat[,1],CN.dat[,2])
title(main="Standard Errors - Ordinary Kriging")
```



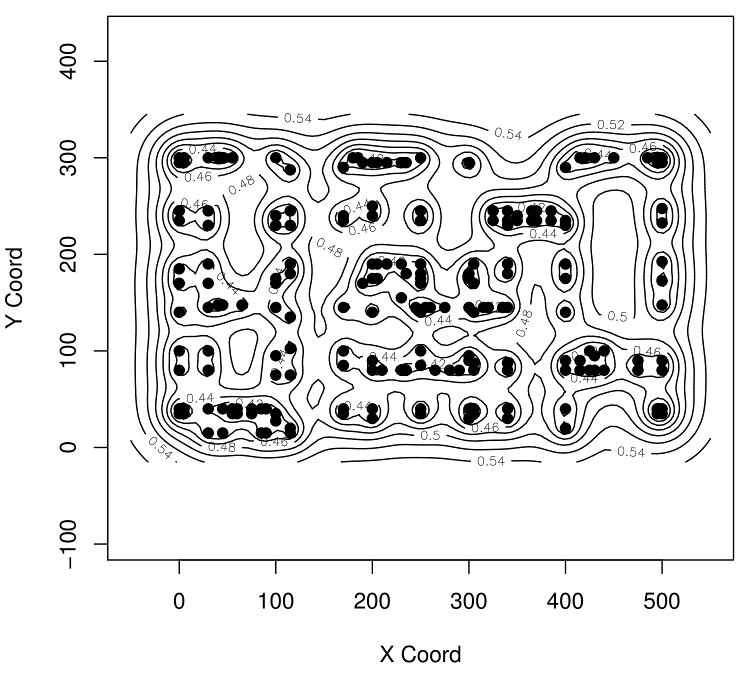
You can also look at perspective plots or contour plots.

```
contour(CN.krige)
points(CN.dat[,1],CN.dat[,2],pch=16)
title(main="Predicted Values - Ordinary Kriging")
contour(CN.krige,val=sqrt(CN.krige$krige.var))
points(CN.dat[,1],CN.dat[,2],pch=16)
title(main="Standard Errors - Ordinary Kriging")
```

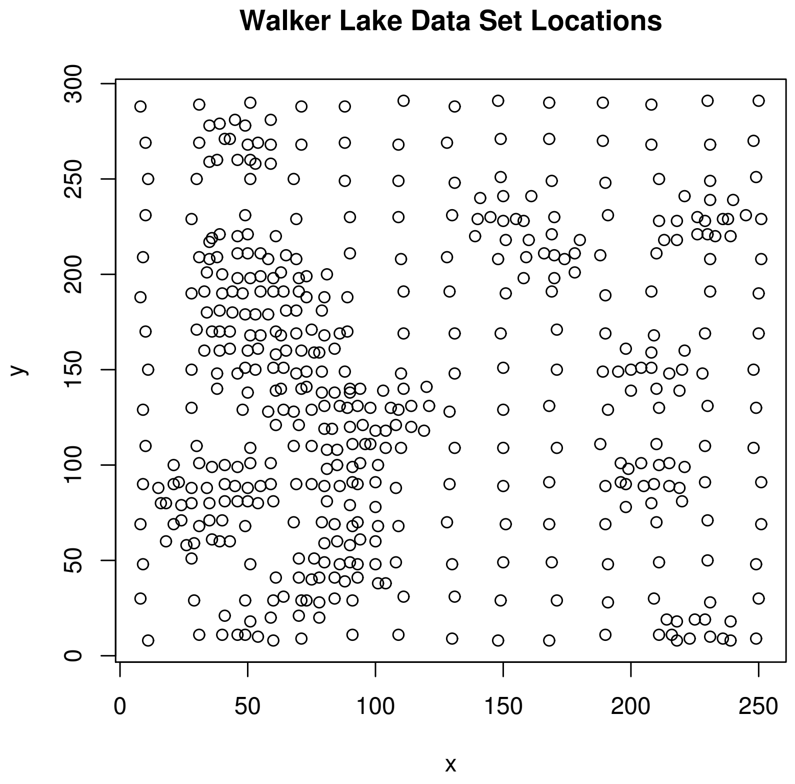
Predicted Values – Ordinary Kriging



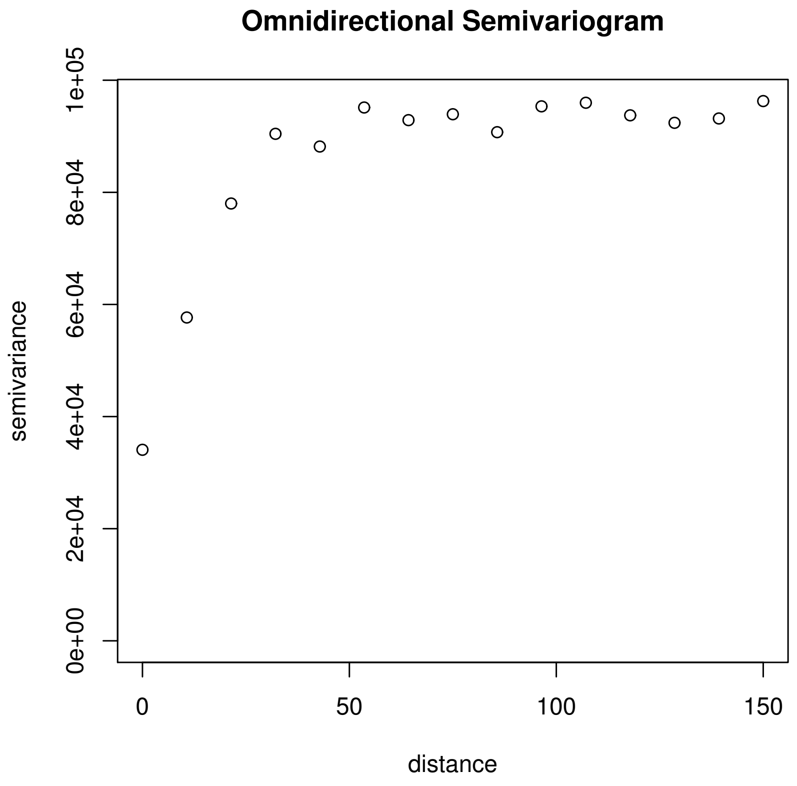
Standard Errors – Ordinary Kriging



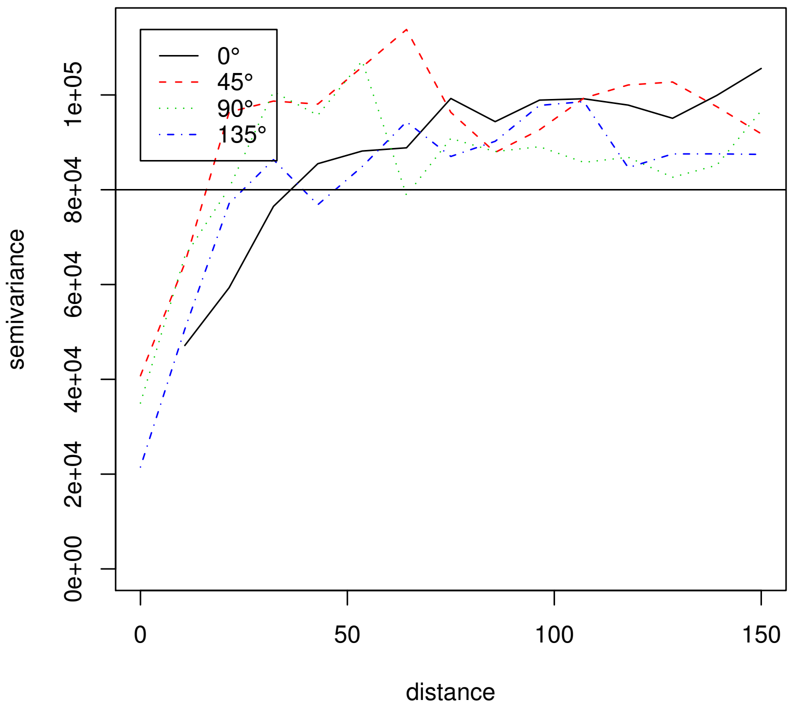
- We will look at an example of ordinary kriging with an anisotropic semivariogram. The data are from Isaaks and Srivastava (the Walker Lake data set). We will consider a continuous variable V which could be taken to be the level of some contaminant in soil samples. This example will cover some of the material we learned in Chapter 4 along with the kriging results from Chapter 5. The locations of the samples are shown below.



The values of V range from 0 to 1528 and the lag distances range from 2m to 370.4m. It is generally advisable to fit an isotropic semivariogram first, even when anisotropy is suspected. The isotropic semivariogram will be based on the largest sample size and should produce the clearest indication of overall structure, e.g. sill, range, etc. The plot below shows an isotropic sample semivariogram.

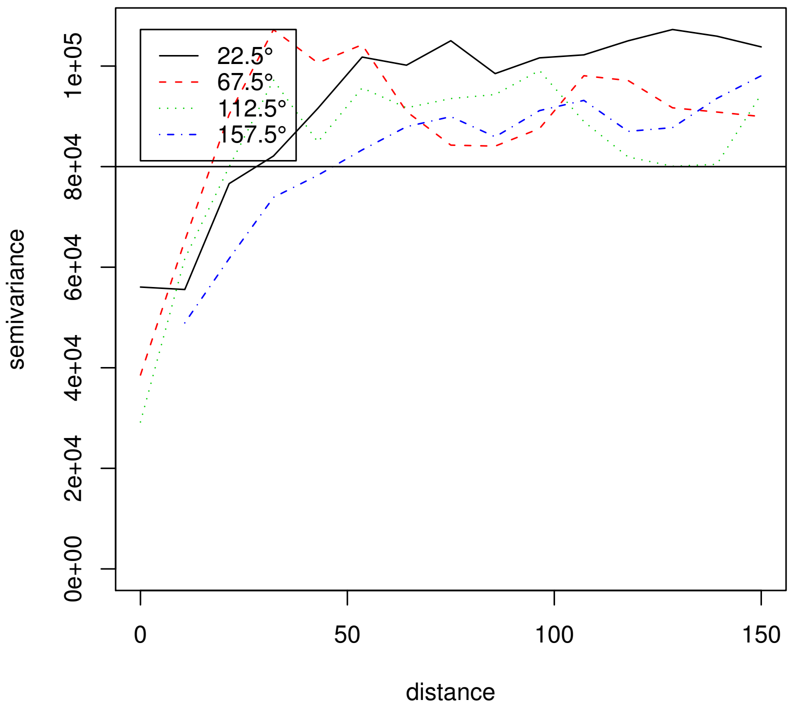


The plot shows a fairly well-behaved semivariogram with indications of a nugget effect of around maybe $30,000$, a sill of around $90,000$, and a range around 45 to 50 meters. Identifying the major and minor axes of anisotropy can be labor intensive. Below is a plot of sample semivariograms computed in the 4 default directions by `variog4`. Those 4 directions are $0°, 45°, 90°, 135°$. Also shown is a horizontal line at $80,000$. The horizontal line intersects the $0°$ semivariogram at about 40m. The $45°$ and $90°$ semivariograms intersect the horizontal line around 20m. This suggests a major axis trending north and a minor axis oriented in a NE-SW direction. We can try to nail it down a bit more precisely by fitting another set of 4 directional semivariograms.



A second directional plot is shown below.

```
walker.variog4<-variog4(walker.geodat,uvec=seq(0,150,l=15),
 direction=rad.fun(c(22.5,67.5,112.5,157.5)))
plot(walker.variog4)
```



The `variog4` function requires directions in radians. I wrote a simple function to do the conversion although there is one in `R` (I think).

```
rad.fun<-function(x){x/360*2*pi}
```

I then constructed sample semivariograms for the 4 directions 22.5°, 67.5°, 112.5°, 157.5°.

The maximum "range" is in the direction of 157.5° and the minimum is in the direction of 67.5°. We could continue to explore these directions in an effort to refine our axes but we will assume that these 2 directions do define the major and minor axes of anisotropy. There is some variability in the directional sills but we will ignore that. I fit empirical semivariograms using `variofit`. It does not appear that it is possible to fit anisotropic semivariograms using `likfit`.

```
variofit(walker.var.omni,ini.cov.pars=c(70000,40),fix.nugget=F,
 nugget=20000,cov.model="spherical",weights="cressie")

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

```
variofit: model parameters estimated by WLS (weighted least squares):
covariance model is: spherical
parameter estimates:
     tausq    sigmasq        phi
26785.1901 66823.4141    38.0255
Practical Range with cor=0.05 for asymptotic range: 38.02548

walker.var.major<-walker.variog4$"157.5"
walker.var.minor<-walker.variog4$"67.5"

## I ran into a problem trying to fit the model in the major axis
## direction
variofit(walker.var.major,ini.cov.pars=c(70000,40),fix.nugget=F,
 nugget=20000,cov.model="spherical",weights="cressie")
variofit(walker.var.major,ini.cov.pars=c(70000,40),fix.nugget=F,
+  nugget=20000,cov.model="spherical",weights="cressie")
variofit: covariance model used is spherical
variofit: weights used: cressie
variofit: minimisation function used: optim
Error in if (ini.cov.pars[1] > 2 * vmax) warning("unreasonable initial value for si
  missing value where TRUE/FALSE needed

## There is a missing value in the empirical semivariogram estimate
walker.var.major$v
 [1]       NA 48922.59 61702.76 73881.62 78277.61 83320.13 87934.26 89945.58 85919.
 [10] 91142.37 93169.55  87013.67 87755.60 93565.52 98067.16
## We could try some heroics but based on plots and results in the
## other empirical fits I just "eyeballed" a value of 40000 for the
## missing value.
walker.var.major$v[1]<-40000
variofit(walker.var.major,ini.cov.pars=c(70000,40),fix.nugget=F,
  nugget=20000,cov.model="spherical",weights="cressie")
variofit: covariance model used is spherical
variofit: weights used: cressie
variofit: minimisation function used: optim
variofit: model parameters estimated by WLS (weighted least squares):
covariance model is: spherical
parameter estimates:
     tausq    sigmasq        phi
40246.1104 50399.4286    75.3143
Practical Range with cor=0.05 for asymptotic range: 75.31432
```

```
variofit(walker.var.minor,ini.cov.pars=c(70000,40),fix.nugget=F,
 nugget=20000,cov.model="spherical",weights="cressie")

variofit: weights used: cressie
variofit: minimisation function used: optim
variofit: model parameters estimated by WLS
(weighted least squares):
covariance model is: spherical

parameter estimates:
     tausq     sigmasq       phi
37372.3653  56438.7838   29.0637
```

The results are summarized (with some obvious roundoff) below.

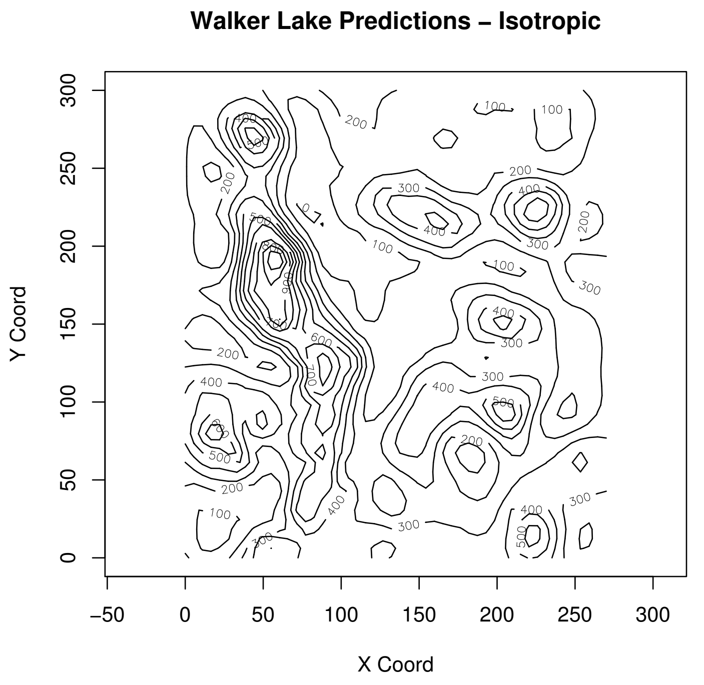
	Nugget	Partial Sill	Range
Isotropic	27,000	67,000	38
Major Axis	40,000	50,000	75
Minor Axis	37,000	56,500	29

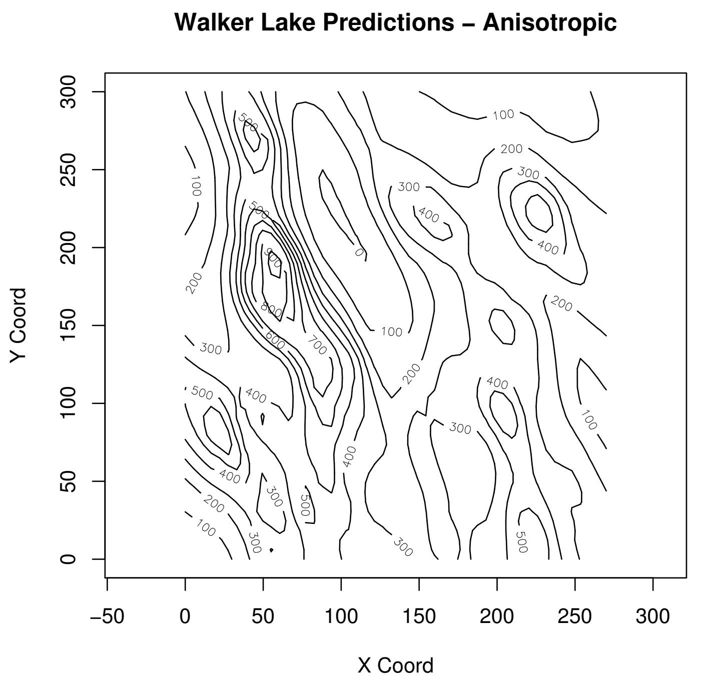
The nugget effects are all comparable as are the partial sills. The ranges differ quite a bit. I am going to krige using a geometrically anisotropic semivariogram with nugget effect of 36,000, sill of 90,000, an anisotropic ratio of 75/29, and an anisotropy angle of 157.5°.

```
walker.grid<-expand.grid(seq(0,270,len=50),seq(0,300,len=50))
walker.kr<-krige.conv(walker.geodat,locations=walker.grid,
    krige=krige.control(cov.model="spherical",cov.pars=c(54000,75),
    nugget=36000,aniso.pars=c(rad.fun(157.5),75/29)))
```

I also kriged assuming isotropy.

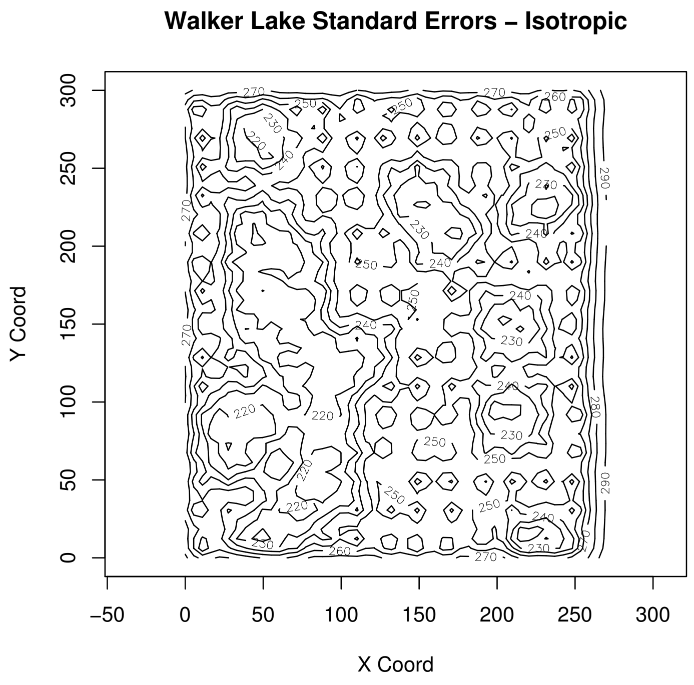
```
walker.kr.iso<-krige.conv(walker.geodat,locations=walker.grid,
       krige=krige.control(cov.model="spherical",cov.pars=c(54000,40),
       nugget=34000))
```

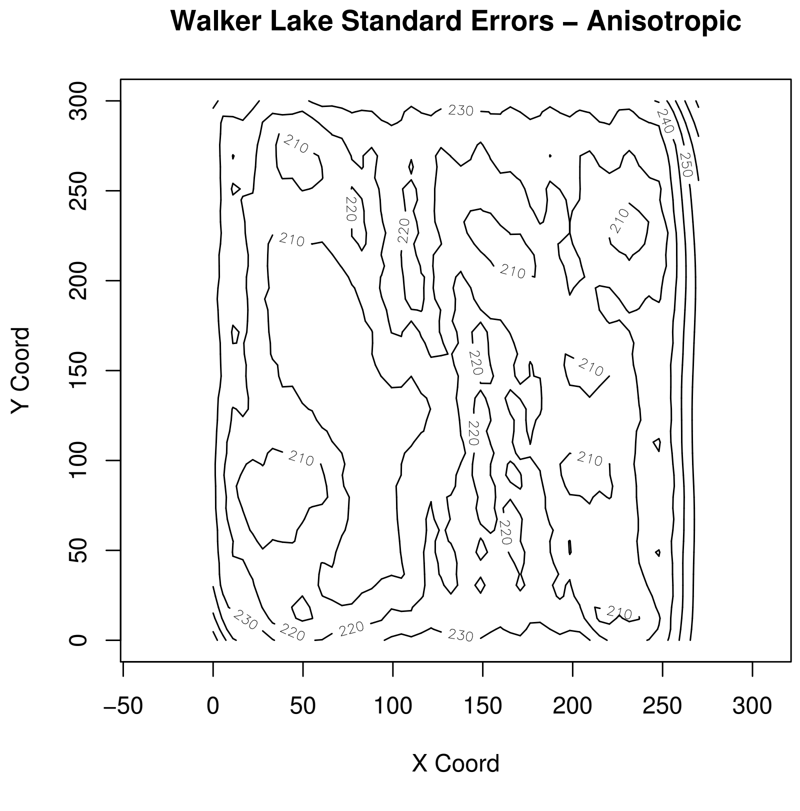




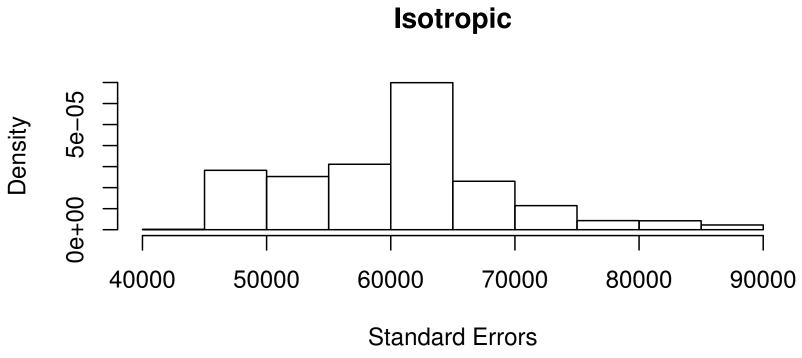
Note the effect of the anisotropy.

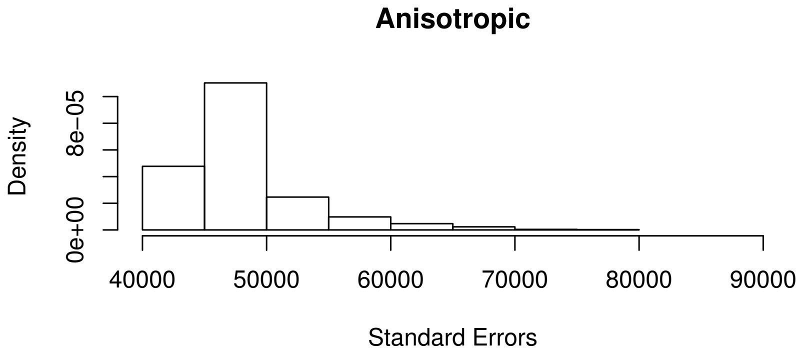
The standard error plots are shown below.


Walker Lake Standard Errors – Isotropic
Y Coord
X Coord
0
50
100
150
200
250
300
–50
220
230
240
250
260
270
280
290


Walker Lake Standard Errors – Anisotropic
Y Coord
X Coord
0
50
100
150
200
250
300
–50
210
220
230
240
250

The kriging standard errors are less for the anisotropic version than for the isotropic version.

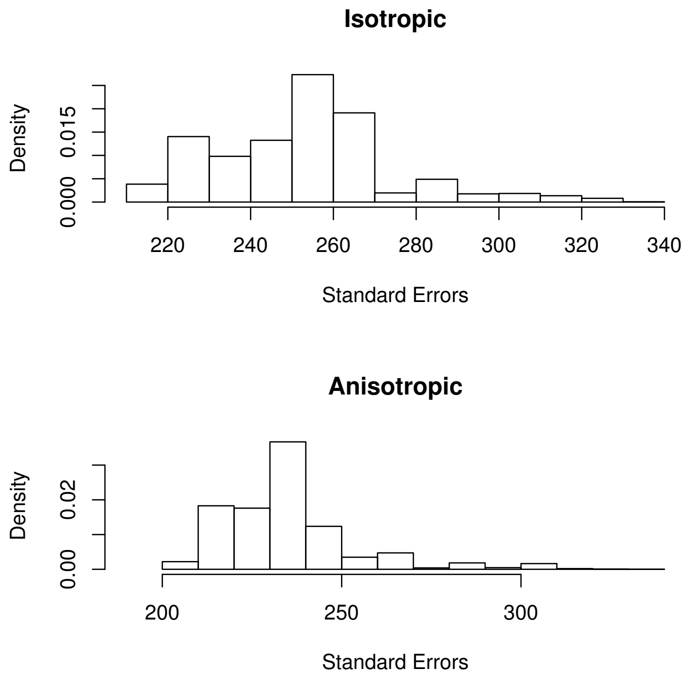


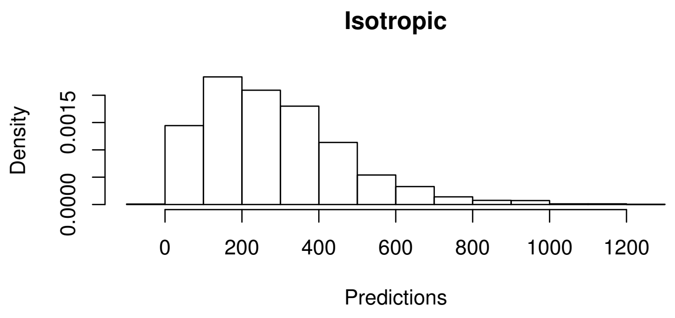


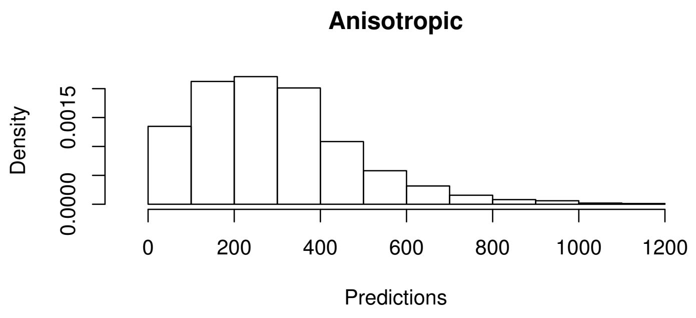
The predictions are roughly comparable. The median of the ratio of the predictions is 1.02. However, there are some atypical locations where the 2 approaches yield very different results. For example, the predicted value at coordinate $(71.6, 226.5)$ is -1.51 for the isotropic approach and 142.47 for the anisotropic version. Almost all of the anomalous values occur in the northwest part of the sampled region where levels of V at the sampled locations are changing rapidly. For example, the observed value of V at coordinate $(71, 160)$ is 801.1 but it drops to 0 at coordinate $(71, 268)$. In fact, a high value of 1280 and a low of 0 occur within 20 meters of each other in this general area. Anisotropic and isotropic semivariograms can easily yield different predictions in such a rapidly changing zone of the study area.

Another potential problem is that the `krig.conv` kriges globally, i.e. all locations are used in computing the predicted value. Geostatisticians generally like to limit the number of points used in predictions. There are a number of reasons for this. For one thing the assumption of mean stationarity is more tenable over local regions. Another reason is because of the behavior we have just seen where Z values are changing rapidly over short distances. There is a function `ksline` which will do such local prediction. It is not particularly well documented. In this case it seems to produce better results.

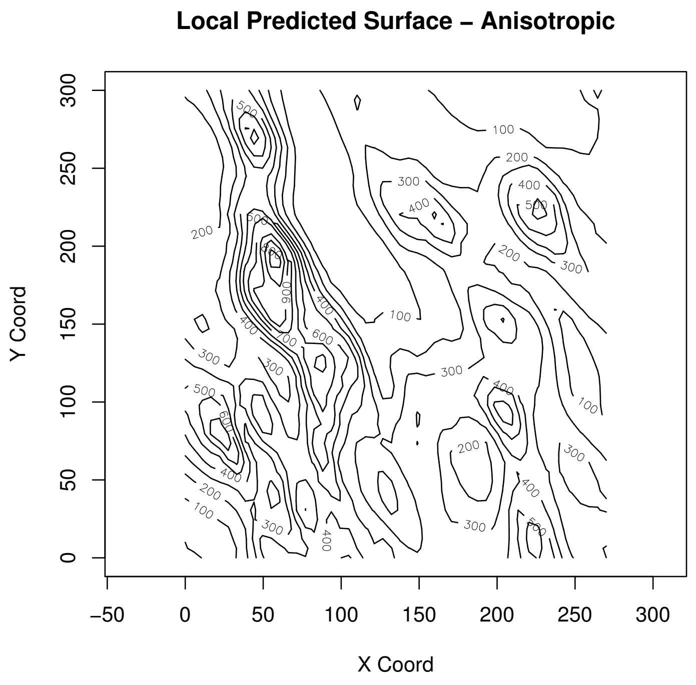
```
walker.loc.anis<-ksline(walker.geodat,locations=walker.grid,nwin=12,
 cov.model="spherical",cov.pars=c(60000,40),nugget=34000,
 aniso.pars=c(rad.fun(157.5),75/29))
walker.loc<-ksline(walker.geodat,locations=walker.grid,nwin=12,
 cov.model="spherical",cov.pars=c(60000,40),nugget=34000)}
```



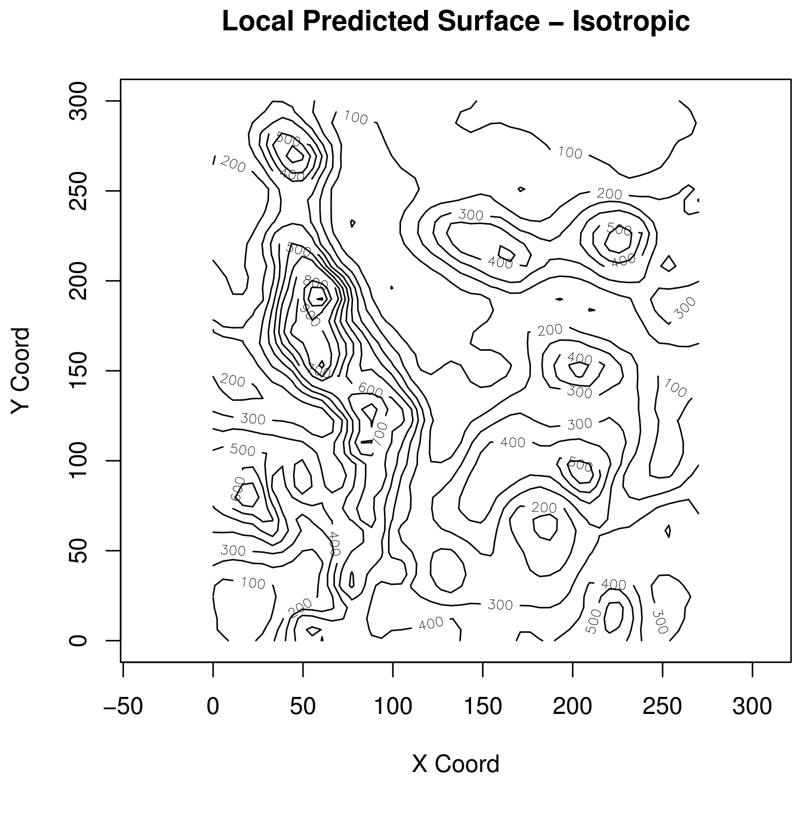




The standard errors from the anisotropic approach tend to be smaller than those computed assuming isotropy. Note also that we do not get any impossible negative predictions from the anisotropic version. Contour plots of the predictions are shown below.



Local Predicted Surface – Anisotropic
Y Coord
X Coord
0
50
100
150
200
250
300
–50



Local Predicted Surface – Isotropic
Y Coord
X Coord
0
50
100
150
200
250
300
–50

The predicted value at $(71.6, 226.5)$ is 40.72 assuming isotropy and 67.86 assuming anisotropy.

Spatial Prediction with a Spatially Varying Mean

We have considered simple kriging and ordinary kriging. The model of the spatial process we have been working with is $\mathbf{Z}(\mathbf{s}) = \boldsymbol{\mu}(\mathbf{s}) + \mathbf{e}(\mathbf{s})$ where $\mathbf{e}(\mathbf{s}) \sim (\mathbf{0}, \boldsymbol{\Sigma})$. Simple kriging assumes that both $\boldsymbol{\mu}(\mathbf{s})$ and $\boldsymbol{\Sigma}$ are known (note that simple kriging does not assume a constant mean function). Ordinary kriging assumes a constant unknown mean function, i.e. $\boldsymbol{\mu}(\mathbf{s}) = \mu$ for all $\mathbf{s}$, and a known $\boldsymbol{\Sigma}$. Assumptions of a known spatially varying mean function or an unknown but constant mean function are often unrealistic. It seems likely that the mean will depend on $\mathbf{s}$,

$$E\left[Z\left(\mathbf{s}\right)\right] = \mu\left(\mathbf{s}\right).$$

Universal Kriging

- We assume
$$\mathbf{Z}\left(\mathbf{s}\right) = \mathbf{X}\left(\mathbf{s}\right)\boldsymbol{\beta} + \mathbf{e}\left(\mathbf{s}\right)$$
where $\boldsymbol{\beta}$ is an unknown vector of parameters and $\mathbf{e}\left(\mathbf{s}\right)$ is as described above. That is we assume that the mean of Z is a linear function of spatially varying predictors. Theoretically, these predictors could be any characteristic of a sampled location. The mean structure $\mathbf{X}\left(\mathbf{s}\right)\boldsymbol{\beta}$ is generally assumed to model larger scale variability (spatial trends or *drift* as geostatisticians refer to it).

- We desire prediction of Z at a new location $\mathbf{s}_0$ and assume that the predicted value is a linear combination of the sampled values
$$p\left(\mathbf{Z};\mathbf{s}_0\right) = \sum_{i=1}^{n} a_i Z\left(\mathbf{s}_i\right) = \mathbf{a}'\mathbf{Z}\left(\mathbf{s}\right).$$

- We seek weights a_i to minimize the mean squared prediction error
$$E\left\{\left[p\left(\mathbf{Z};\mathbf{s}_0\right) - Z\left(\mathbf{s}_0\right)\right]^2\right\}$$
subject to a constraint that the predictor is unbiased in the sense that
$$E\left[p\left(\mathbf{Z};\mathbf{s}_0\right)\right] = E\left[Z\left(\mathbf{s}_0\right)\right] = \mathbf{x}\left(\mathbf{s}_0\right)'\boldsymbol{\beta}$$
where $\mathbf{x}\left(s_0\right)$ is a vector of predictors at $\mathbf{s}_0$.

- It can be shown that
$$E\left\{\left[p\left(\mathbf{Z};\mathbf{s}_0\right) - Z\left(\mathbf{s}_0\right)\right]^2\right\} = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a} + C\left(\mathbf{0}\right) - 2\mathbf{a}'\boldsymbol{\sigma}$$
where $\boldsymbol{\Sigma}$ is the variance-covariance matrix of the observed values, $C\left(\mathbf{0}\right) = \text{Var}\left(Z\left(\mathbf{s}_0\right)\right)$, and $\boldsymbol{\sigma} = \text{Cov}\left(\mathbf{Z}\left(\mathbf{s}\right), Z\left(\mathbf{s}_0\right)\right)$. It can also be shown (we will skip the details this time-see pages 241-242 for an outline of the steps) that the solution is
$$\mathbf{a} = \boldsymbol{\Sigma}^{-1}\left[\boldsymbol{\sigma} - \mathbf{X}\left(\mathbf{s}\right)\left(\mathbf{X}\left(\mathbf{s}\right)'\boldsymbol{\Sigma}^{-1}\mathbf{X}\left(\mathbf{s}\right)\right)^{-1}\left(\mathbf{X}\left(\mathbf{s}\right)'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} - \mathbf{x}\left(\mathbf{s}_0\right)\right)\right].$$

The $BLUP$ is then

$$\begin{aligned}
\mathbf{a}'\mathbf{Z}(\mathbf{s}) &= \left[\boldsymbol{\sigma} - \mathbf{X}(\mathbf{s})\left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} - \mathbf{x}(\mathbf{s}_0)\right)\right]'\boldsymbol{\Sigma}^{-1}\mathbf{Z}(\mathbf{s}) \\
&= \left[\boldsymbol{\sigma} - \mathbf{X}(\mathbf{s})\left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} + \mathbf{X}(\mathbf{s})\left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\mathbf{x}(\mathbf{s}_0)\right]' \\
&\quad \times \ \boldsymbol{\Sigma}^{-1}\mathbf{Z}(\mathbf{s}) \\
&= \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\mathbf{Z}(\mathbf{s}) - \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\widehat{\boldsymbol{\beta}}_{gls} + \mathbf{x}(\mathbf{s}_0)'\widehat{\boldsymbol{\beta}}_{gls} \\
&= \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\left(\mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\widehat{\boldsymbol{\beta}}_{gls}\right) + \mathbf{x}(\mathbf{s}_0)'\widehat{\boldsymbol{\beta}}_{gls}
\end{aligned}$$

where

$$\widehat{\boldsymbol{\beta}}_{gls} = \left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{Z}(\mathbf{s}).$$

The mean squared prediction error (minimized) is

$$\begin{aligned}
\sigma^2_{uk}(\mathbf{s}_0) &= C(\mathbf{0}) - \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} \\
&\quad + \left(\mathbf{x}(\mathbf{s}_0)' - \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)\left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)^{-1} \\
&\quad \times \ \left(\mathbf{x}(\mathbf{s}_0)' - \boldsymbol{\sigma}'\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{s})\right)'
\end{aligned}$$

These can be written much more simply in terms of the a_i's as

$$\begin{aligned}
p_{uk}(\mathbf{Z};\mathbf{s}_0) &= \sum_{i=1}^{n} a_i Z(\mathbf{s}_i) \\
\sigma^2_{uk}(\mathbf{s}_0) &= C(\mathbf{0}) - 2\sum_{i=1}^{n} a_i\sigma_{i0} + \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \sigma_{ij}
\end{aligned}$$

where $\sigma_{i0} = \mathrm{Cov}(Z(\mathbf{s}_i), Z(\mathbf{s}_0))$ and $\sigma_{ij} = \mathrm{Cov}(Z(\mathbf{s}_i), Z(\mathbf{s}_j))$.

- Universal kriging is often used to fit polynomial trend models with mean

$$\mu(\mathbf{s}) = \mu(x,y) = \sum_{k=0}^{p}\sum_{m=0}^{p}\beta_{km}x^k y^m.$$

For example, a quadratic trend surface could be modeled using the mean function

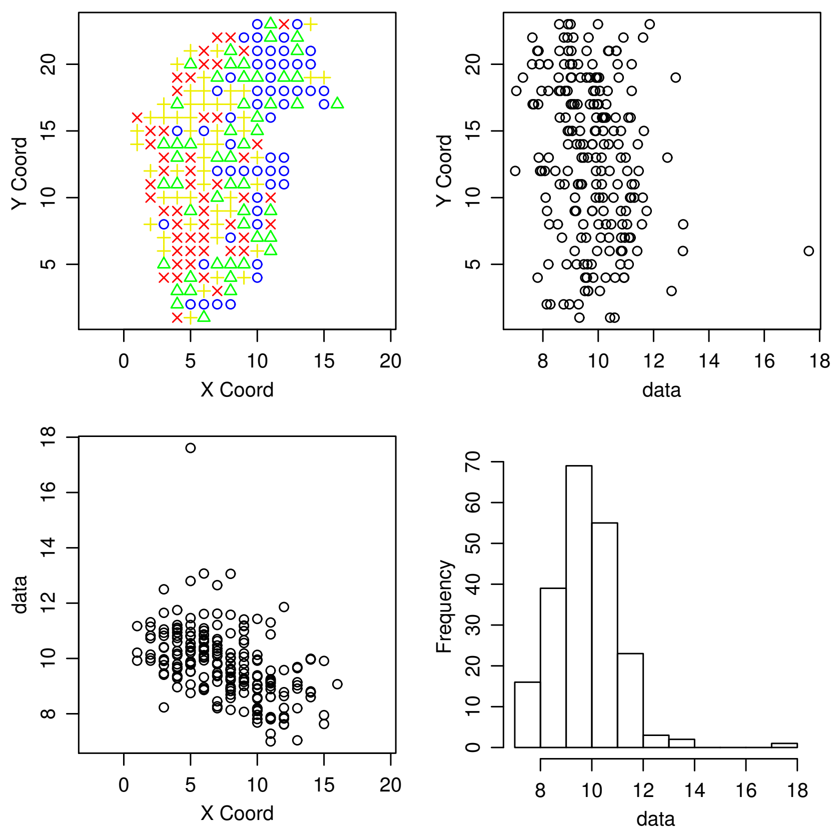
$$\mu(x,y) = \beta_{00} + \beta_{10}x + \beta_{01}y + \beta_{11}xy + \beta_{20}x^2 + \beta_{02}y^2.$$

There is nothing in the theory however that limits us to predictors that are powers of the spatial coordinates. We could include any number of variables that characterize a particular location (e.g. elevation, aspect, etc.).

- *A simple example: This example will be rather long winded but hopefully helpful. The data are 208 core measurements of coal ash in a seam of coal in Pennsylvania and are in a* `geodata` *object in R;* `coal.geodata`.

  Cressie (1993) has made this data set famous and you can find an extensive analysis of it in his text. A set of summary plots is shown below produced by the command

```
plot(coal.geodata)}
```



Note that there appears to be a linear trend in the east-west direction but no trend in the north-south direction. Note the appearance of an unusual value. Assuming a linear trend in the east-west direction a plausible model would be

$$Z(\mathbf{s}) = \beta_{00} + \beta_{10}x + e(\mathbf{s})$$

where $e(\mathbf{s})$ *is a zero mean second order stationary error process.*

Estimation of the semivariogram is problematical in the presence of trend. You showed on an earlier homework that the empirical semivariogram estimator is biased in the presence of trend. We have a couple of ways to proceed, both of which have problems of their own.

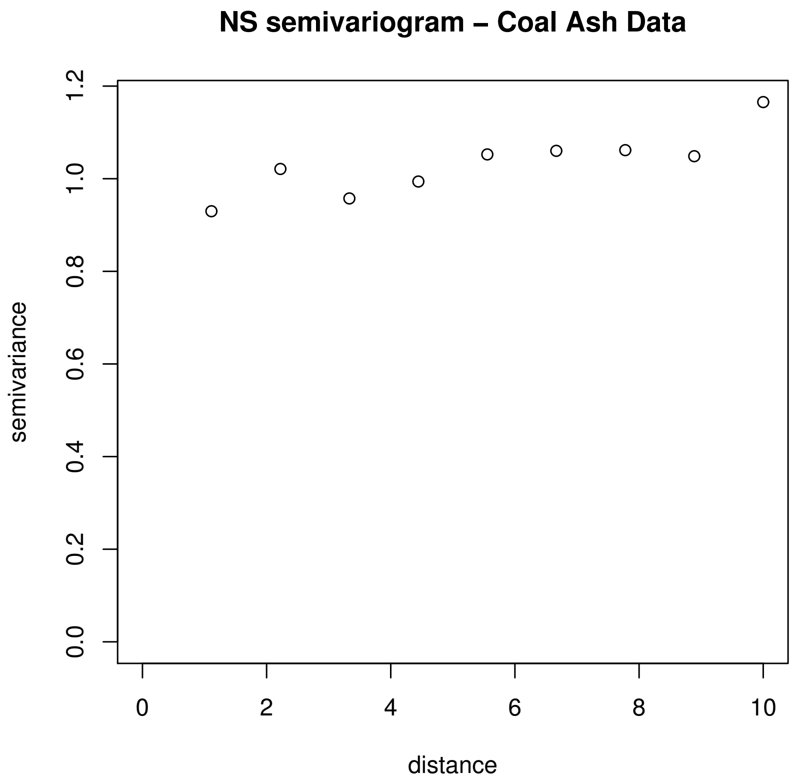
- *We can assume isotropy, fit a semivariogram in the north-south (trendless) direction and use that as our semivariogram.*
- *We could detrend the surface and fit a semivariogram to the residuals.*

We will consider method 1 in this example and look at detrending in more detail below. The plot below shows a north-south semivariogram. A skeptic might conclude that this is a pure nugget effect model but that would make this example uninteresting so I am going to use the model Cressie fit, which is a spherical model with nugget of 0.89, partial sill of 0.14, and range of 4.31.

I produced the plot with the commands

```
coal.var<-(variog(coal.geodat,uvec=seq(0,10,l=10),estimator="modulus",
 tolerance=rad.fun(20),direction=0))
variog: computing variogram for direction = 0 degrees (0
radians) tolerance angle = 20 degrees (0.349 radians)
plot(coal.var)
```

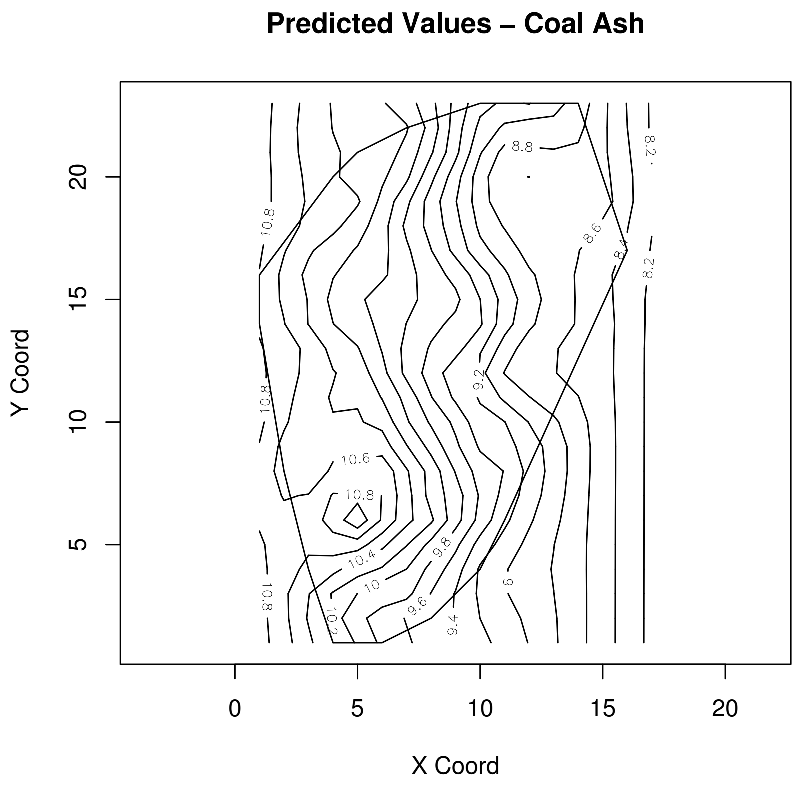
Note the `estimator="modulus"` *argument. This results in the robust empirical semivariogram. This would seem prudent given the unusual observation.*



The region is not a rectangle but an irregular polygon. We need to keep that in mind when viewing the results.

The first contour plot below shows prediction from global kriging. I used the `ksline` *function which presumably is less efficient but ran fast enough for this exercise.*

```
coal.grid<-expand.grid(seq(1,17,1),seq(1,23,1))
coal.kr.global<-ksline(coal.geodat,locations=coal.grid,
 cov.model="spherical",cov.pars=c(0.14,4.32),nugget=0.89,
       m0="kt",trend=1)
contour(coal.kr.global,nlevels=15)
lines(coal.dat[hpts,1],coal.dat[hpts,2])
```

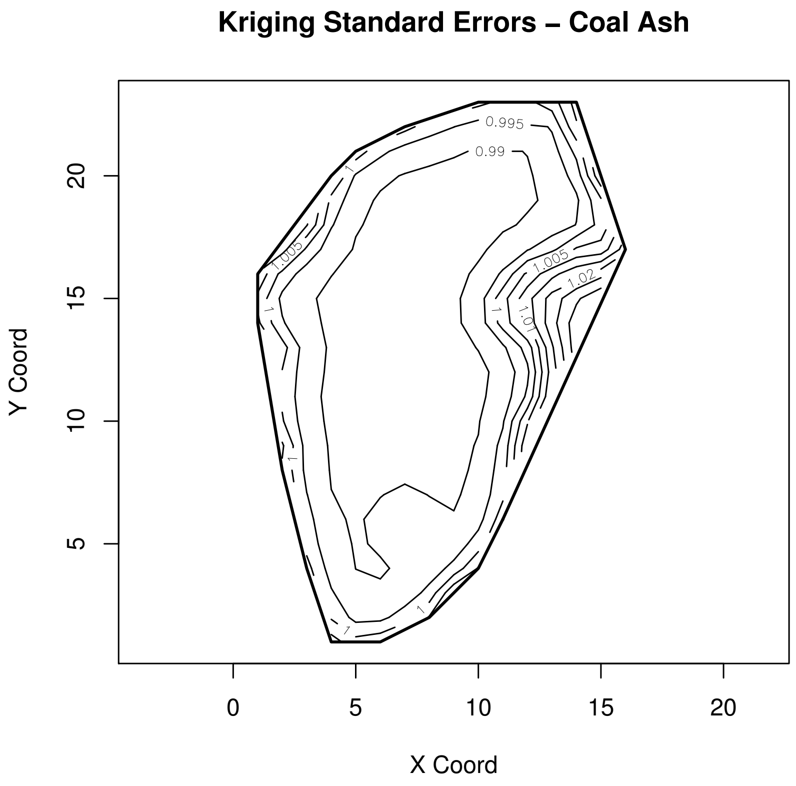


The `lines` *command drew in the convex hull with* `hpts` *containing the rows of the points at the vertices of the hull. Those were identified using the commands below.*

```
hpts<-chull(coal.dat[,1],coal.dat[,2])
hpts<-c(hpts,hpts[1])}
```

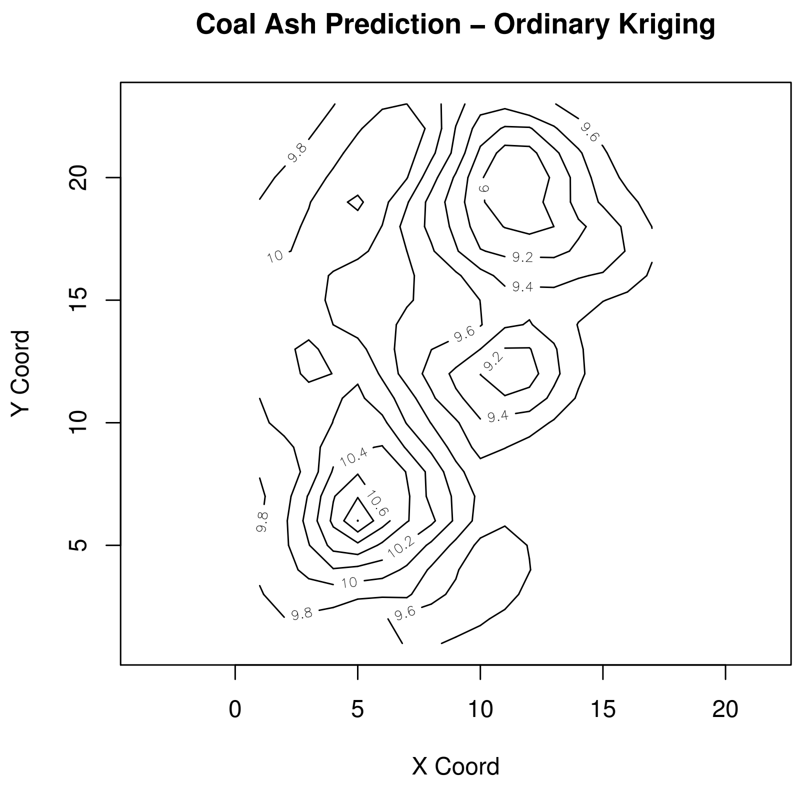
A plot of the kriging standard errors is somewhat uninformative because the larger standard errors outside the hull swamp information about the values inside the hull. Further they are fairly constant inside the hull. We can do a better job on focusing our attention inside the irregular polygon in which the data were measured using the `borders` *argument. First store the coordinates at the vertices of the polygon in an object. Then krige within that region.*

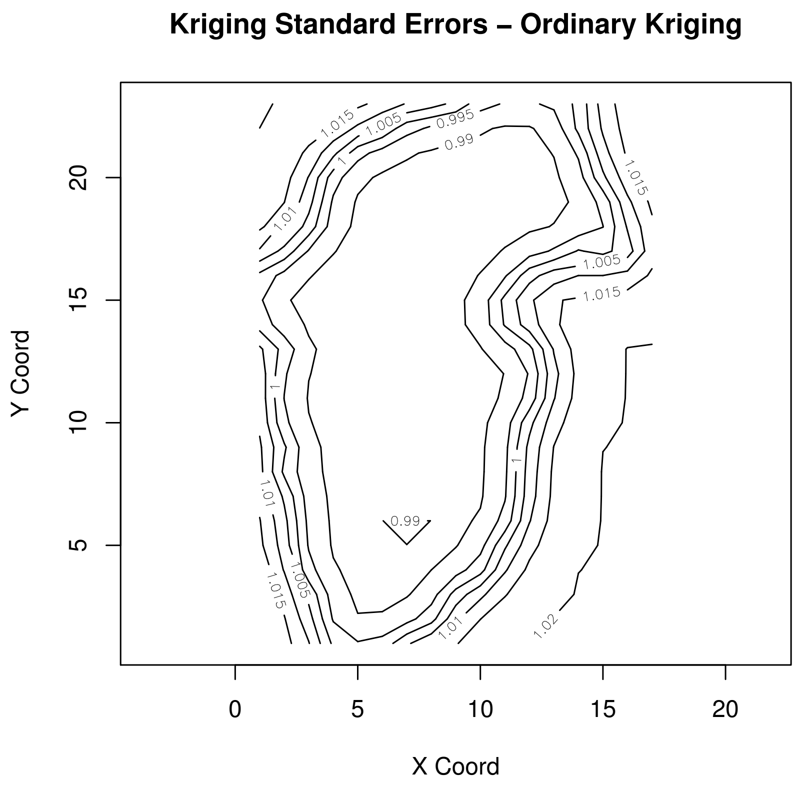
```
bord<-coal.dat[chull(coal.dat[,1],coal.dat[,2]),-3]
coal.kr.border<-ksline(coal.geodat,locations=coal.grid,borders=bord,
 cov.model="spherical",cov.pars=c(0.14,4.32),nugget=0.89,
 m0="kt",trend=1)
ksline:results will be returned only for prediction
locations inside the borders

contour(coal.kr.border,val=sqrt(coal.kr.border$krige.var))
title(main="Kriging Standard Errors - Coal Ash")}
```



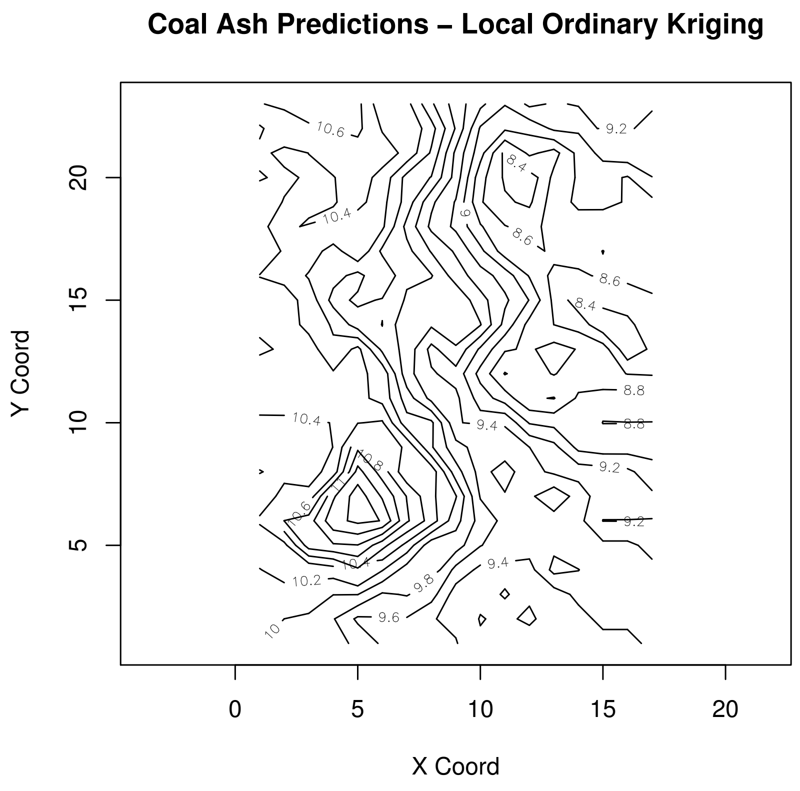
You can see that the standard errors are fairly constant.

What if we ignored the trend? The contour plot below shows predictions from ordinary kriging.


Coal Ash Prediction – Ordinary Kriging
Y Coord
X Coord
0
5
10
15
20
5
10
15
20
9.8
10
9.6
9
9.2
9.4
9.6
9.2
9.4
10.4
10.6
9.8
10.2
10
9.8
9.6



The predictions do seem to be different. The global ordinary kriging does not seem to have picked up the trend as well. What if we krige locally? The results from a local ordinary kriging are shown below. They seem to match the universal kriging results a bit better.



It would be nice to have a method for comparing results in a more objective manner. Cross-validation offers one option and we will look at that now.

An adequate semivariogram should produce good predictions in that $p(\mathbf{Z};\mathbf{s}_0)$ should be close to $Z(\mathbf{s}_0)$. If we were doing a standard regression analysis we could compare predicted values with observed values but that will not work with kriging because it is an exact interpolator. Predictions could be "ground-truthed" to see how good they are. This may not be feasible. Data could be held out of the model fitting phase and predictions for the held out locations could be checked against the truth which is known. However, as a general rule, data are hard to come by and scientists like to use all of it in analysis. Cressie suggests the following approach.

- *Fit a semivariogram model using all of the data.*
- *Delete an observation $Z(\mathbf{s}_j)$ and predict it with $p_{-j}(\mathbf{Z};\mathbf{s}_j)$. One also gets a cross-validated kriging variance, $\sigma^2_{-j}(\mathbf{s}_j)$.*
- *Characterize the closeness of the prediction with the "truth". Two methods he proposes are*

  1.

$$CV_1 = \frac{1}{n}\sum_{j=1}^{n}\left\{\frac{(Z(\mathbf{s}_j) - p_{-j}(\mathbf{Z};\mathbf{s}_j))}{\sigma_{-j}(\mathbf{s}_j)}\right\}$$

2.

$$CV_2 = \left[\frac{1}{n} \sum_{j=1}^{n} \left\{ \frac{\left(Z\left(\mathbf{s}_j\right) - p_{-j}\left(\mathbf{Z}; \mathbf{s}_j\right)\right)}{\sigma_{-j}\left(\mathbf{s}_j\right)} \right\}^2 \right]^{1/2}$$

CV_2 is related to the PRESS statistic you may have seen in applied regression courses. We will consider it below.

This is fairly easy to program in R. An example of the code I used to cross-validate the universal kriging model is shown below.

```
for(i in 1:208){
a<-ksline(coords=coal.dat[-i,1:2],data=coal.dat[-i,3],
                locations=coal.dat[i,1:2], cov.model="spherical",
                 cov.pars=c(0.14,4.31),nugget=0.89,m0="kt")
coal.xval.preduk[i]<-a$predict
coal.xval.seuk[i]<-a$krige.var}}
```

It is worth remembering the unusual value. It may well skew the cross-validation results. The table below shows CV_2 for 4 models: global ordinary kriging, local ordinary kriging, global universal kriging, and local universal kriging for all 208 data points and for 207 with the unusual observation deleted.

Kriging Model	*CV_2 (all data)*	*CV_2 (outlier deleted)*
Ordinary (Global)	1.146	1.032
Ordinary (Local)	1.095	0.972
Universal (Global)	1.109	0.991
Universal (Local)	1.070	0.943

Ordinary global kriging performs the worst. The other 3 produce comparable results when all data are used, but local universal kriging does not perform that well when the outlier is deleted. Overall it appears that local ordinary or global universal are comparable regardless of whether or not the outlier is deleted.

- We have seen some methods for estimation of covariance (semivariogram) parameters in Chapter 4. These methods are applicable under the assumption of intrinsic stationarity. We will look at 2 topics here:

  1. Estimation of covariance parameters when there is a spatially varying mean.
  2. The impact of estimation on prediction.

- Much of the material in the text requires more background in the theory of linear models than expected. I am going to take a low key approach and try to give us a feel for the basic results.

- *Iteratively Reweighted Generalized Least Squares*: We have the spatial linear model:

$$\mathbf{Z}(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \mathbf{e}(\mathbf{s})$$

  where $\mathbf{e}(\mathbf{s}) \sim (\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$. Recall that

$$p_{uk}(\mathbf{Z}; \mathbf{s}_0) = \mathbf{x}(\mathbf{s}_0)' \widehat{\boldsymbol{\beta}}_{gls} + \boldsymbol{\sigma}(\boldsymbol{\theta})' \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} \left(\mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\widehat{\boldsymbol{\beta}}_{gls}\right)$$

  with

$$\widehat{\boldsymbol{\beta}}_{gls} = \left(\mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} \mathbf{Z}.$$

  We must estimate $\boldsymbol{\theta}$, the parameters of the semivariogram (covariance function). You showed in the last homework that for all pairs of coordinates $(\mathbf{s}_i, \mathbf{s}_j)$ such that $\mathbf{h} = \| \mathbf{s}_i - \mathbf{s}_j \|$

$$E\left[\widehat{\gamma}_Z(\mathbf{h})\right] = \gamma_e(\mathbf{h}) + \frac{1}{2|N(\mathbf{h})|} \sum_{N(\mathbf{h})} \left\{\mathbf{X}(s_i)\boldsymbol{\beta} - \mathbf{X}(\mathbf{s}_j)\boldsymbol{\beta}\right\}^2,$$

  i.e. you showed that estimating $\boldsymbol{\theta}$ in this way will yield biased results (and badly biased results at that). The method of *estimated generalized least squares* is often used to estimate $\boldsymbol{\beta}$. The method works as follows (page 257 in our text).

  1. Estimate $\boldsymbol{\beta}$ by the method of least squares:

$$\widehat{\boldsymbol{\beta}}_{ols} = \widehat{\boldsymbol{\beta}}_{(0)} = \left(\mathbf{X}(\mathbf{s})' \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \mathbf{Z}(\mathbf{s}).$$

  2. Obtain the residuals

$$\mathbf{r} = \mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\widehat{\boldsymbol{\beta}}_{ols}.$$

  3. Estimate γ_e (and thus $\boldsymbol{\theta}$) using the residuals and one of the methods we discussed earlier in Chapter 4 (OLS, WLS, MLE, $REML$). Denote this estimator as $\widehat{\boldsymbol{\theta}}_{(0)}$.

4. Reestimate $\boldsymbol{\beta}$:

$$\widehat{\boldsymbol{\beta}}_{(1)} = \left(\mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{(0)}\right)^{-1} \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{(0)}\right)^{-1} \mathbf{Z}$$

5. Steps $2-4$ are iterated until the estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ stabilize. This usually does not take many iterations.

Denote the final estimate of $\boldsymbol{\theta}$ by $\widehat{\boldsymbol{\theta}}_{(\cdot)}$. Then the estimated generalized estimator of $\boldsymbol{\beta}$ is

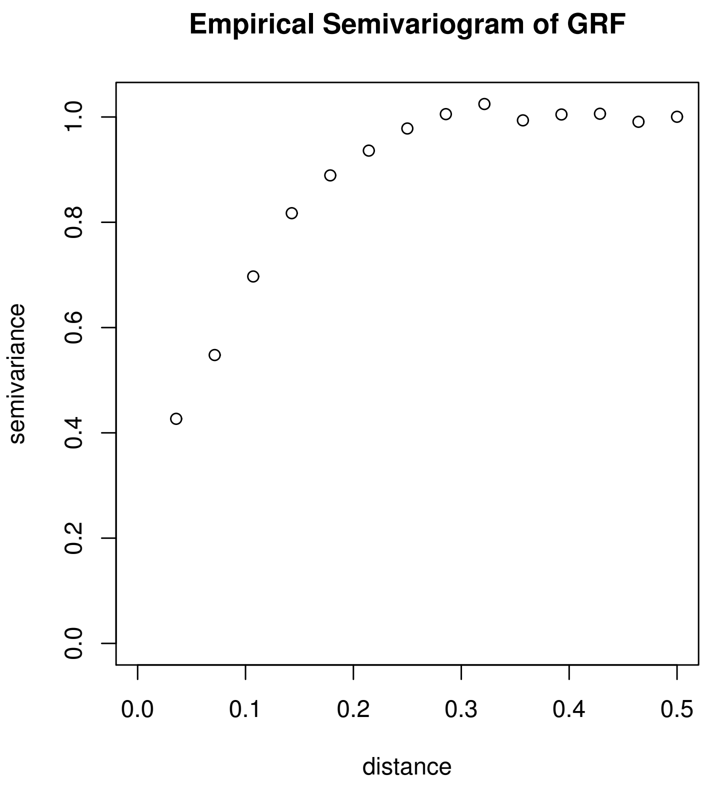
$$\widehat{\boldsymbol{\beta}}_{egls} = \left(\mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{(\cdot)}\right)^{-1} \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{(\cdot)}\right)^{-1} \mathbf{Z}.$$

One problem with this approach is that the estimator of the semivariogram is still (negatively) biased. The bias increases with increasing lag and Cressie recommends the WLS method of semivariogram estimation using the weights given in equation 4.34 on page 165 of our text as a method of controlling for the bias. These weights are available in `variofit` by specifying `weights="cressie"` in the argument of `variofit`. The method just described is referred to as ***I****teratively* ***R****e****W****eighted* ***G****eneralized* ***L****east* ***S****quares* or $IRWGLS$. The text denotes the resulting estimator as $\widehat{\boldsymbol{\theta}}_{IRWGLS}$ but we will use the more convenient $\widehat{\boldsymbol{\theta}}_{(\cdot)}$. Generally, the procedure converges quickly and 2 or 3 iterations are generally enough.

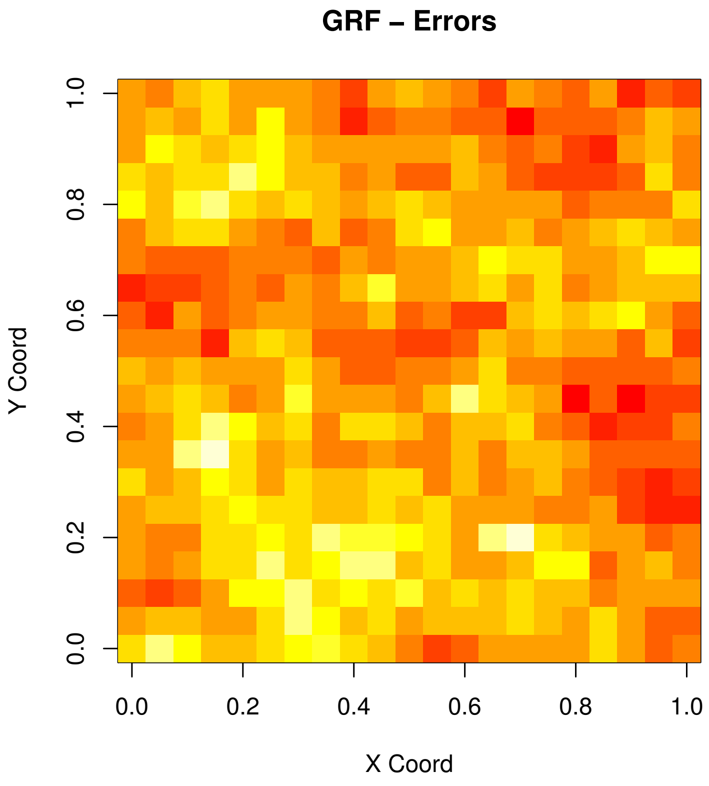
- *A Simple Example: Simulate a Gaussian random field using the* `grf` *function in* `geoR`*. The covariance function is chosen to be an exponential model with nugget effect of 0, effective range of 0.25, and sill of 1. I will assume that the nugget effect is known to be 0.*

```
grf.ex<-grf(441,grid="reg",cov.model="exponential",cov.pars=c(1,0.25/3))
```

This simulates a Gaussian random field over a regular 21×21 *grid. Below is a plot of the semivariogram,* $\widehat{\gamma}_e$*. I fit an exponential semivariogram using* WLS *and estimated the sill to be* 1.015 *and the effective range to be* 0.2544 *so the covariance structure of this simulated data set is very close to the true covariance structure.*



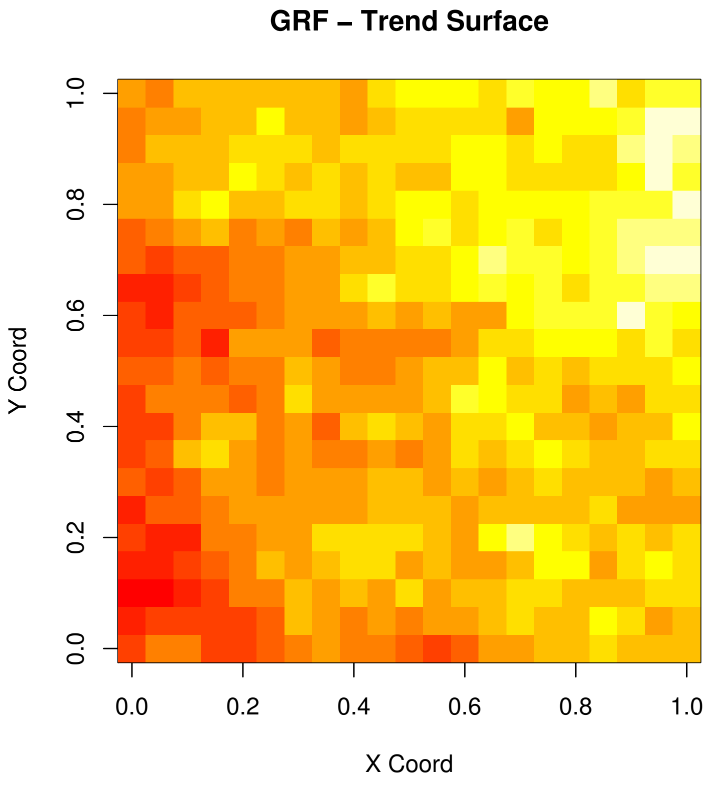
The random field itself is shown below.



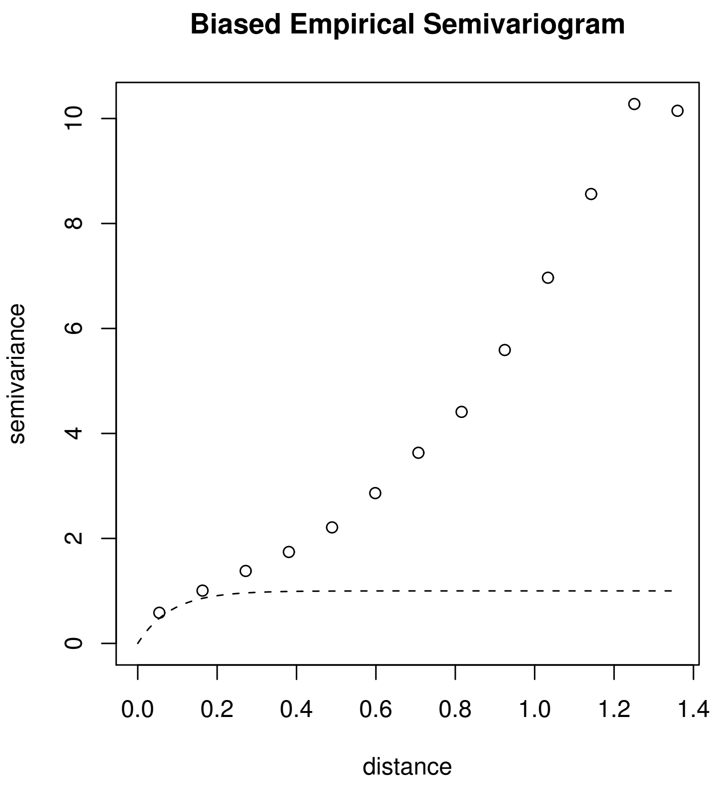
I then added a trend structure as follows. The coordinates of the random field are in `grf.ex$coords` *and the data can be found in* `grf.ex$data`*.*

```
grf.ex.trend<-grf.ex
grf.ex.trend$data<-1+5*grf.ex$coords[,1]+3*grf.ex$coords[,2]+grf.ex$data
```

The result is shown below.



The resulting trend is hopefully pretty obvious. The empirical semivariogram computed on these trended Z values is shown below along with the true semivariogram of the process. Clearly the results are badly biased. Note that the empirical semivariogram looks like a power model. There are some who suggest that the appearance of such an empirical semivariogram where you did not expect to see one suggests an unsuspected trend.

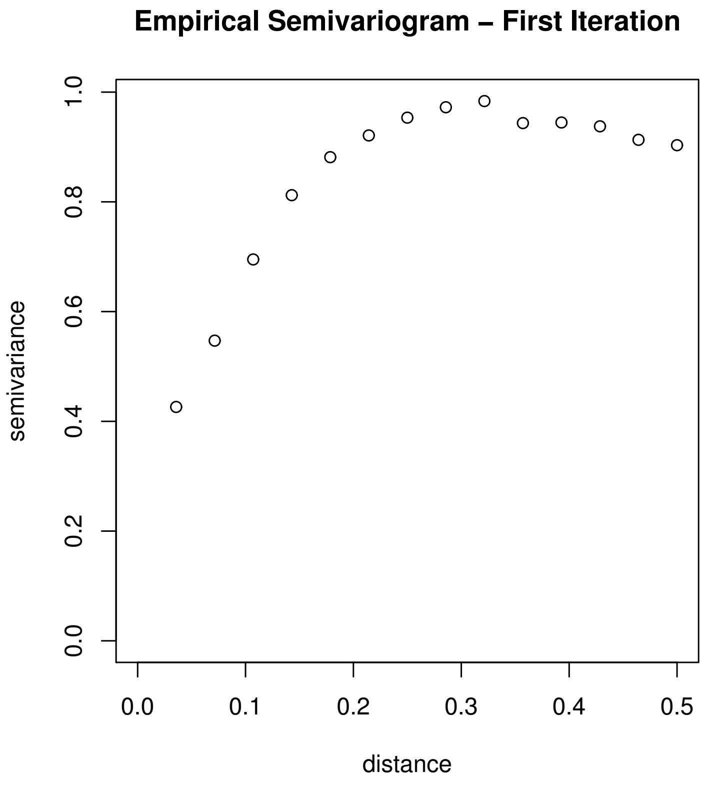


We can fit a trend surface to the data easily because the `lm` *function recognizes* `grf.ex.trend` *as a data frame.*

```
trend.lm<-lm(data~x+y,data=grf.ex.trend)
coef(trend.lm)
 (Intercept)     x         y
   1.973054  4.107095  2.242615
```

You can compare the fitted values to the true values above.

Now we fit a semivariogram model to the empirical semivariogram of the residuals (shown below). The parameter estimates were: sill= 0.947 *and effective range*= 0.206*.*



In terms of the notation from above we now have $\widehat{\boldsymbol{\beta}}_{(0)} = [1.97, 4.11, 2.24]^{'}$ *and* $\widehat{\boldsymbol{\theta}}_{(0)} = [0.947, 0.206]^{'}$.

Automatic iteration of the procedure is a bit of a problem because we really should look at the semivariogram to get reasonable starting values for the ***variofit*** *function. I will give it a try without specifying those, i.e. I will let* ***variofit*** *choose the starting values.*

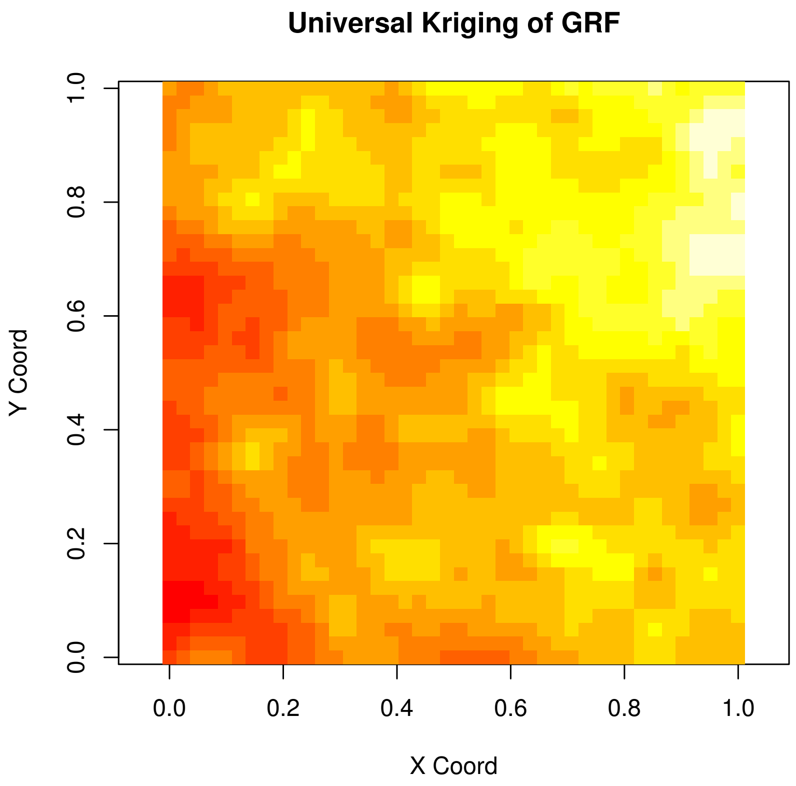
I wrote the following function.

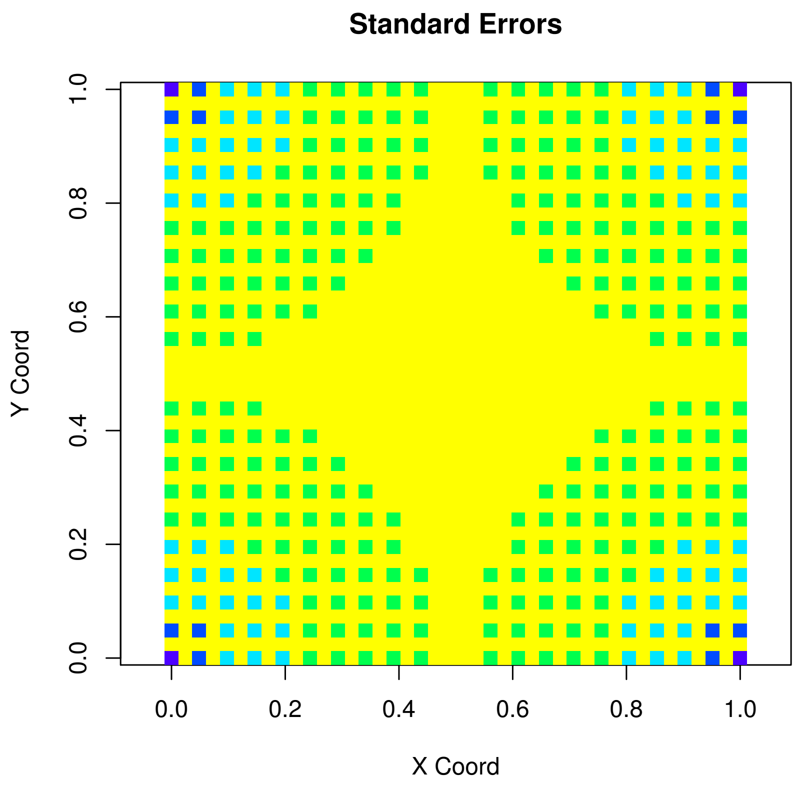
```
irwgls<-function(sill,p.range,trend.X,trend.Z,trend.dist){
Sig.inv<-solve(cov.spatial(trend.dist,cov.model="exponential",
               cov.pars=c(sill,p.range)))
bhat<-solve(t(trend.X)%*%Sig.inv%*%trend.X)%*%
 t(trend.X)%*%Sig.inv%*%trend.Z
resid<-trend.Z-trend.X%*%bhat
a<-variog(coords=grf.ex.trend$coords,data=resid,uvec=seq(0,.5,l=15))
theta.hat<-variofit(a,ini.cov.pars=c(sill,p.range/3),
  fix.nugget=T,cov.model="exponential",weights="cressie")
cov.parms<-theta.hat$cov.pars
return(cov.parms)}
```

I ran 5 iterations starting with the $\widehat{\boldsymbol{\theta}}_{(0)}$ *values from above and got the following results.*

Iteration	*Sill*	*effective range*
1	0.9475343	0.2068239
2	0.94677215	0.2064432
3	0.94677143	0.2064428
4	0.94677143	0.2064428
5	0.94677143	0.2064428

The estimates stabilized very quickly. An image of the fitted surface (with predictions made over a 42×42 *grid) is seen below.*





Although they are not of primary concern in our example the $\boldsymbol{\beta}$ *estimates also stabilized quickly. These coefficients will be of more interest in Chapter 6 when we discuss spatial regression models.*

- *Maximum Likelihood Estimation*: This is always a good choice *assuming you can live with the assumptions one of which is that the data are truly from a Gaussian random field*. The objective function (minus twice the log likelihood) is a function of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ and is given (equation 5.41 on page 259) as

$$\varphi(\boldsymbol{\beta};\boldsymbol{\theta};\mathbf{Z}(\mathbf{s})) = \ln\{|\boldsymbol{\Sigma}(\boldsymbol{\theta})|\} + n\ln(2\pi) + (\mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\boldsymbol{\beta})'\,\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}(\mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\boldsymbol{\beta}).$$

The number of parameters that must be estimated is $k+q$ where $k = \text{rank}(\mathbf{X})$ and q is the number of covariance parameters. The text describes a method of reducing the number of parameters to be estimated by *profiling*, a procedure that leads to a different objective function (see equation 5.44 on page 260). The procedure involves 6 steps:

1. Set $\partial\varphi/\partial\beta_j = 0$ for $j = 1, \cdots, p$ and solve. The solution is

$$\widehat{\boldsymbol{\beta}}_{gls} = \left(\mathbf{X}(\mathbf{s})'\,\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\mathbf{X}(\mathbf{s})'\,\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{Z}(\mathbf{s}).$$

2. Write $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \sigma^2 \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)$. $\boldsymbol{\theta}^*$ is a $(q-1) \times 1$ vector. This is usually possible with most commonly used covariance models because they have corresponding correlation models that are proportional to the covariance models in which case σ^2 is the proportionality constant and $\boldsymbol{\Sigma}(\boldsymbol{\theta}^*)$ is the correlation matrix. Given this we can rewrite

$$\widehat{\boldsymbol{\beta}}_{gls} = \left(\mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)^{-1} \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)^{-1} \mathbf{Z}(\mathbf{s})$$

because σ^2 cancels.

3. Substitute these expressions into φ yielding

$$\varphi_{\boldsymbol{\beta}}(\boldsymbol{\theta}; \mathbf{Z}(\mathbf{s})) = \ln\left\{|\sigma^2 \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)|\right\} + n\ln(2\pi) + \sigma^{-2}\mathbf{r}' \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)^{-1}\mathbf{r}.$$

The vector $\mathbf{r}$ is the residual

$$\mathbf{r} = \mathbf{Z}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\widehat{\boldsymbol{\beta}}_{gls}.$$

4. Find the MLE of σ^2 as a function of $\boldsymbol{\theta}^*$. The solution is

$$\widehat{\sigma}^2_{ml} = \frac{1}{n}\mathbf{r}' \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)^{-1}\mathbf{r}.$$

5. Substitute this into $\varphi_{\boldsymbol{\beta}}$ to get

$$\varphi_{\boldsymbol{\beta},\sigma}(\boldsymbol{\theta}^*; \mathbf{Z}(\mathbf{s})) = \ln\{|\boldsymbol{\Sigma}(\boldsymbol{\theta}^*)|\} + n\ln\left(\widehat{\sigma}^2_{ml}\right) + n(\ln(2\pi) - 1)$$

and set derivatives with respect to the elements of $\boldsymbol{\theta}^*$ equal to 0. Solving yields $\widehat{\boldsymbol{\theta}}^*_{ml}$.

6. Finally, find $\widehat{\boldsymbol{\theta}}_{ml}$ from $\widehat{\sigma}^2_{ml}$ and $\widehat{\boldsymbol{\theta}}^*_{ml}$ and find

$$\widehat{\boldsymbol{\beta}}_{ml} = \left(\mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{ml}\right)^{-1} \mathbf{X}(\mathbf{s})\right)^{-1} \mathbf{X}(\mathbf{s})' \boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{ml}\right)^{-1} \mathbf{Z}(\mathbf{s}).$$

Fortunately, the `likfit` function in `geoR` will do this for us.

GRF example-continued: We have the simulated Gaussian random field from above. We need starting values for our covariance parameters. The WLS solutions are good enough for that. You could also just eyeball them from $\widehat{\gamma}$.

```
grf.ex.mle<-likfit(grf.ex.trend,trend="1st",ini.cov.pars=c(0.947,0.206/3),
 fix.nugget=T,cov.model="exponential")
```

The solutions are

```
likfit: estimated model parameters:

    beta0    beta1    beta2   sill    phi
   1.8392   4.1430   2.3226 0.8982 0.0791
```

The MLE for the effective range is 0.2373*.*

- *REML Estimation*: REML or Restricted Maximum Likelihood is a method of finding estimators that have better bias properties than MLEs. The text points out, for example, that with and *iid* sample from a $N(\mu, \sigma^2)$ the MLE of σ^2 is

$$\widehat{\sigma}^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2$$

This is know to be biased low for σ^2. The unbiased estimator generally preferred is the REML Estimator

$$\widehat{\sigma}^2_{reml} = \frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2$$

REML estimation requires a transformation of $\mathbf{Z}(\mathbf{s})$ $(\mathbf{KZ}(\mathbf{s}))$ with mean $\mathbf{0}$. The objective function is

$$\varphi(\boldsymbol{\theta}; \mathbf{Z}(\mathbf{s})) = \ln\left\{|\mathbf{K}\boldsymbol{\Sigma}(\boldsymbol{\theta})\mathbf{K}'|\right\} + (n-k)\ln(2\pi) + \mathbf{Z}(\mathbf{s})'\mathbf{K}'\left(\mathbf{K}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{K}'\right)^{-1}\mathbf{KZ}(\mathbf{s}).$$

Note that $\boldsymbol{\beta}$ is seemingly gone. A *REML* estimator of $\boldsymbol{\beta}$ does not exist. It is estimated via the method of Estimated Generalized Least Squares as

$$\widehat{\boldsymbol{\beta}}_{egls} = \left(\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{reml}\right)^{-1}\mathbf{X}(\mathbf{s})\right)^{-1}\mathbf{X}(\mathbf{s})'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}_{reml}\right)^{-1}\mathbf{Z}(\mathbf{s}).$$

More details are given on pages 261-263 including information on the mysterious $\mathbf{K}$. We will skip those here.

GRF example - continued: We can use `likfit` *to get REML estimates.*

```
grf.ex.reml<-likfit(grf.ex.trend,trend="1st",ini.cov.pars=c(0.947,0.206/3),
 fix.nugget=T, cov.model="exponential",method="REML")
```

The solution is

```
likfit: estimated model parameters:
   beta0    beta1    beta2   sill      phi
  1.8297   4.1430   2.3263 0.9621    0.0862
```

The estimate of the effective range is 0.2586.

- When the assumption of a Gaussian random field is plausible likelihood based methods are preferred over *IRWGLS*. Recall (page 47) that a spatial process $Z(\mathbf{s})$ is a Gaussian random field if $\mathbf{Z}(\mathbf{s}) = \left[Z(\mathbf{s}_1), Z(\mathbf{s}_2), \cdots, Z(\mathbf{s}_k)\right]'$ is multivariate normal for all k. This is a nontrivial assumption and likelihood methods should not be used without adequate justification.

- *Estimation of Covariance Parameters and Prediction Errors*: Our predictions and the estimates of the uncertainty in those predictions as quantified by the kriging variance have assumed that the covariance model and its parameters are known. Section 5.5.4 contains a brief discussion of the effect of estimation of the parameters on prediction errors. *Note that this section still assumes that the covariance (semivariogram) model is known.*

  Consider the ordinary kriging predictor,

$$p_{ok}(\mathbf{Z}; \mathbf{s}_0) = \left[\boldsymbol{\sigma}(\boldsymbol{\theta}) + \mathbf{1}\left(\frac{1 - \mathbf{1}'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\boldsymbol{\sigma}(\boldsymbol{\theta})}{\mathbf{1}'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{1}}\right)\right]\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{Z}(\mathbf{s}).$$

  The *plug-in* estimator of this is

$$\widehat{p}_{ok}(\mathbf{Z}; \mathbf{s}_0) = \left[\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\theta}}\right) + \mathbf{1}\left(\frac{1 - \mathbf{1}'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\theta}}\right)}{\mathbf{1}'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\mathbf{1}}\right)\right]\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\mathbf{Z}(\mathbf{s}).$$

  Similarly, the kriging variance was given as

$$\sigma^2_{ok} = C(\mathbf{0}) - \boldsymbol{\sigma}(\boldsymbol{\theta})'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\boldsymbol{\sigma}(\boldsymbol{\theta}) + \frac{\left(1 - \mathbf{1}'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\boldsymbol{\sigma}(\boldsymbol{\theta})\right)^2}{\mathbf{1}'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{1}}$$

  and its plug-in estimator is

$$\widehat{\sigma}^2_{ok} = C(\mathbf{0}) - \boldsymbol{\sigma}\left(\widehat{\boldsymbol{\theta}}\right)'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\theta}}\right) + \frac{\left(1 - \mathbf{1}'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\boldsymbol{\sigma}\left(\widehat{\boldsymbol{\theta}}\right)\right)^2}{\mathbf{1}'\boldsymbol{\Sigma}\left(\widehat{\boldsymbol{\theta}}\right)^{-1}\mathbf{1}}.$$

  There are 2 (at least) implications of this:

  1. $\widehat{p}_{ok}$ is not the *BLUP*; it is an estimator of the *BLUP* or *EBLUP*.
  2. $\widehat{\sigma}^2_{ok}$ is a biased estimator of the kriging variance of p_{ok}, not an estimator of the variance of $\widehat{p}_{ok}$.

Assuming $\boldsymbol{\theta}$ is known the mean squared error of p_{ok} is

$$\mathrm{MSE}\left(p_{ok}\left(\mathbf{Z};\mathbf{s}_0\right), Z\left(\mathbf{s}_0\right), \boldsymbol{\theta}\right) = E\left[\left(p_{ok}\left(\mathbf{Z},\mathbf{s}_0\right) - Z\left(\mathbf{s}_0\right)\right)^2\right].$$

The mean squared error of $\widehat{p}_{ok}$ is

$$\mathrm{MSE}\left(\widehat{p}_{ok}\left(\mathbf{Z};\mathbf{s}_0\right), Z\left(\mathbf{s}_0\right), \widehat{\boldsymbol{\theta}}\right) = E\left[\left(\widehat{p}_{ok}\left(\mathbf{Z},\mathbf{s}_0\right) - Z\left(\mathbf{s}_0\right)\right)^2\right].$$

It can be shown (under some assumptions-see page 265) that

$$\begin{aligned}\mathrm{MSE}\left(\widehat{p}_{ok}\left(\mathbf{Z};\mathbf{s}_0\right), Z\left(\mathbf{s}_0\right), \widehat{\boldsymbol{\theta}}\right) &= \mathrm{MSE}\left(p_{ok}\left(\mathbf{Z};\mathbf{s}_0\right), Z\left(\mathbf{s}_0\right), \boldsymbol{\theta}\right) + \mathrm{Var}\left[\widehat{p}_{ok}\left(\mathbf{Z};\mathbf{s}_0\right) - p_{ok}\left(\mathbf{Z};\mathbf{s}_0\right)\right] \\ &\geq \mathrm{MSE}\left(p_{ok}\left(\mathbf{Z};\mathbf{s}_0\right), Z\left(\mathbf{s}_0\right), \boldsymbol{\theta}\right)\end{aligned}$$

Correction factors exist but depend on assumptions that may not be reasonable (e.g. GRF). However, after all is said and done the text concludes: "the use of a plug-in estimator of the kriging variance is fine for most spatial problems with moderate to strong spatial autocorrelation."

- Up to now we have been discussing point kriging, i.e. prediction of Z at new point locations. In many applications interest is more appropriately focused on prediction of an average value of Z in a "block" B. For example, a mining engineer may be able to predict the amount of ore at points but the ore will be taken out in larger blocks. This is an example of a *change-of-support* problem, so-called because we are trying to draw inference to a region B based on data collected at points; $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n$.

- We start by assuming a second order stationary process spatial process

$$\{Z(\mathbf{s}) : \mathbf{s} \in D\}$$

with $E[Z(\mathbf{s})] = \mu$ for all $\mathbf{s}$ and covariance function

$$\text{Cov}\,[Z(\mathbf{s}_i), Z(\mathbf{s}_j)] = C(\mathbf{s}_i, \mathbf{s}_j).$$

The data $Z(\mathbf{s}_1), \cdots, Z(\mathbf{s}_n)$ will be used to predict the average value in a block or region B:

$$Z(B) = \frac{1}{|B|} \int_B Z(\mathbf{s}) d\mathbf{s}$$

where $|B|$ is the area or volume of B depending on the dimension of the spatial domain, i.e.

$$|B| = \int_B d\mathbf{s}.$$

B is referred to as the *support* of $Z(B)$.

- We want linear predictors

$$p(\mathbf{Z}; Z(B)) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{s}_i)$$

where the weights λ_i are chosen to minimize the mean-squared prediction error

$$E\left\{[p(\mathbf{Z}; Z(B)) - Z(B)]^2\right\}.$$

The weights are found in a manner exactly analogous to the method used in ordinary point kriging but with one crucial difference. We need to find the covariances between $Z(\mathbf{s}_i), \quad i = 1, \cdots, n$ and $Z(B)$ denoted in our text by

$$\boldsymbol{\sigma}(B, \mathbf{s}) = [\text{Cov}(Z(B), Z(\mathbf{s}_1)), \text{Cov}(Z(B), Z(\mathbf{s}_2)), \cdots, \text{Cov}(Z(B), Z(\mathbf{s}_n))]'$$

with

$$\text{Cov}(Z(B), Z(\mathbf{s})) = \frac{1}{|B|} \int_B C(\mathbf{u}, \mathbf{s}) d\mathbf{u}.$$

- The solution, derived using the method of Lagrange multipliers can be shown to be

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\sigma}(B, \mathbf{s}) + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}(B, \mathbf{s})}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}}\right)' \boldsymbol{\Sigma}^{-1}$$

  with the multiplier m being

$$m = -\frac{1 - \mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}(B, \mathbf{s})}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}}.$$

- The kriging variance is

$$\sigma_{ok}^2 = \sigma(B, B) - \boldsymbol{\lambda}'\boldsymbol{\sigma}(B, \mathbf{s}) - m$$

  where

$$\boldsymbol{\lambda}'\boldsymbol{\sigma}(B, \mathbf{s}) = \boldsymbol{\sigma}(B, \mathbf{s})' \boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}(B, \mathbf{s})$$

  and

$$\sigma(B, B) = \text{Cov}[Z(B), Z(B)] = \frac{1}{|B|^2}\int_B C(\mathbf{u}, \mathbf{v})d\mathbf{u}d\mathbf{v}.$$

- There is nothing new in estimation of C. We use the methods discussed above. The integrals are evaluated by discretizing B into N points $\mathbf{u}_j^*,\ j = 1, \cdots, N$ and approximating the point to block and block to block covariances as

$$\begin{aligned}
\text{Cov}(Z(B), Z(\mathbf{s})) &\approx \frac{1}{N}\sum_{j=1}^{N} C\left(\mathbf{u}_j^*, \mathbf{s}\right) \\
\text{Cov}(Z(B), Z(B)) &\approx \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} C\left(\mathbf{u}_i^*, \mathbf{u}_j^*\right)
\end{aligned}$$

- These results can also be expressed in terms of semivariograms.

- *A Simple Example - Example 5.13: The example in the text is done in terms of the semivariogram. We will look at it in terms of the covariance function. We have a one-dimensional process with point to point covariance function*

$$C(s_i, s_j) = \exp\left(-\frac{3|s_i - s_j|}{5}\right).$$

  The nugget effect is 0*, the sill is* 1 *and the effective range is* 5*. The region B is defined to be the interval $B = (2, 4)$ with $|B| = 2$. If we wished to krige we would need*

$$\sigma(B, s) = Cov(Z(B), Z(s)) = \frac{1}{2}\int_2^4 \exp\left(-\frac{3|u - s|}{5}\right) du.$$

  This is not difficult to integrate as long as you recognize the 3 separate cases that must be considered:

1. $s < 2$

2. $2 \leq s \leq 4$

3. $s > 4$

Suppose we wanted to predict Z at $s = 3$. The solid line shows the point to point covariance function for

$$Cov(Z(3), Z(s_i))$$

for $i = 0, \cdots, 6$. Suppose we wanted to predict $Z((2,4))$. The dashed line shows the point to block covariance function

$$Cov(Z((2,4)), Z(s))$$

for $s \in (0,6)$. As expected

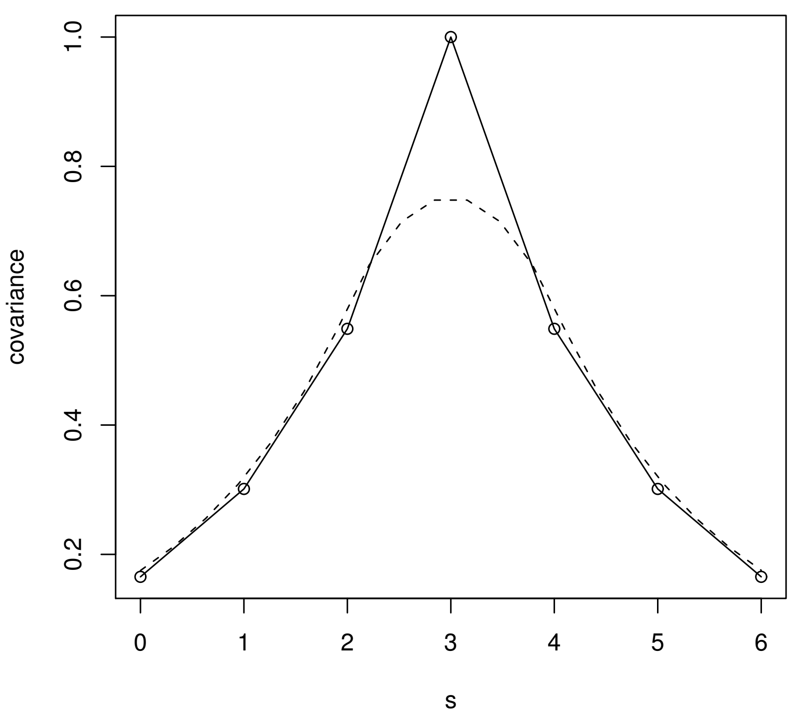
$$Cov(Z(3), Z(3)) = Var(Z(3)) = 1$$

but

$$Cov(Z((2,4)), Z(3)) = 0.752$$

which is the average of the covariances between $Z(3)$ and all other $Z(s)$ values for $s \in (2,4)$.

Example 5.13 – Covariance Function



As a practical matter evaluation of the integrals would be difficult even when the covariance function was assumed to be known. We would often be working in 2 or 3 dimensions on irregularly shaped regions. The approximations described above would typically be used in those cases. We can do that in this simple example, too.

```
# set up a mesh over the interval (2,4)
u<-seq(2,4,l=10000)
# find the mean of the covariances: Cov(Z(u),3)
mean(exp(-3*abs(u-3)/5))
 0.7519603
```

Assume we have observations at locations $s_i = i$, *for* $i = 0, 1, 2, 4, 5, 6$ *and want to use ordinary kriging to predict* $Z(3)$ *and* $Z((2,4))$. *The observations (simulated data from* `grf` *are* $Z(s_0) = 0.164, Z(s_1) = 0.129, Z(s_2) = 0.337, Z(s_4) = 0.217, Z(s_5) = 0.529, Z(s_6) = 0.181$. *The coordinates and data are stored in the 6 by 3 matrix* `coord` *whose first 2 columns contain the locations with the 3rd column containing the data. The prediction at* $Z(s_3)$ *is*

```
z3<-ksline(coords=coord[,1:2],data=coord[,3],locations=c(1,3),
     cov.model="exponential",cov.pars=c(1,15/3))
z3$predict
  0.2756316
z3$krige.var
  0.1976178
```

Now we predict the mean amount in the interval $(2,4)$. *We need the point to block covariances. I "gridded" the interval* $(2,4)$ *into 1000 equal length subintervals and computed the mean of the covariances between the Z values at the 6 locations at which we have observations. For example, the point to block covariance between* $Z(s_0)$ *and* $Z((2,4))$ *is approximated as*

$$Cov(Z((2,4)), Z(s_0)) \approx \frac{1}{1000}\sum_{j=1}^{1000} C(u_j^*, s_0) = 0.175.$$

The R code that did this is

```
u<-seq(2,4,l=1000)
mean(exp(-3*abs(u-0)/5))
0.1754175
```

We do this for all 6 sampled locations. The resulting vector is

$$\boldsymbol{\sigma}(B, s) = [0.1754175 \quad 0.3196315 \quad 0.5824065 \quad 0.5824065 \quad 0.3196315 \quad 0.1754175]'.$$

The variance-covariance matrix (not shown) was generated as follows.

```
d<-as.matrix(dist(coord,upper=T))
Sigmat<-exp(-3*d/5)}
```

The weights are

```
lambda<- t(sigvec.B + one.vec*
 (1 - t(one.vec)%*%solve(Sigmat)%*%sigvec.B)/(t(one.vec)
 %*%solve(Sigmat)%*%one.vec))%*%solve(Sigmat)
lambda
0.02228492 0.01614069 0.4608399 0.457782 0.02425020 0.01870221
```

The predicted mean is

```
sum(lambda*coord[,3])
 0.2765146
```

We need $\sigma(B, B)$ and m (the Lagrange multiplier) for the kriging variance.

```
 m<- -(1 -t(one.vec)%*%solve(Sigmat)%*%sigvec.B)/(
  t(one.vec)%*%solve(Sigmat)%*%one.vec)
# 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
0.6958757
# get the kriging variance
kvar<-0.6958757-(lambda)%*%sigvec.B-m
kvar
0.1795506
```

Note that the block kriging variance is less than the point kriging variance; not unexpected because means are less variable.

Unfortunately, the only package I could find that does block kriging in R is `gstat` *and it is not documented very well. Here is a simple example with many details missing. The data are zinc concentrations measured at 155 locations in a river delta in the Netherlands. The commands below set up the data appropriately for kriging.*

```
require(gstat)
require(sp)
data(meuse)
coordinates(meuse) = ~x+y
data(meuse.grid)
gridded(meuse.grid) = ~x+y
m <- vgm(.59, "Sph", 874, .04)
```

I used the `krige` *command for both point and block ordinary kriging.*

```
# ordinary kriging:
k.point <- krige(log(zinc) ~ 1, meuse, meuse.grid, model = m)
spplot(k.point["var1.pred"], main = "ordinary kriging predictions")
spplot(k.point["var1.var"],  main = "ordinary kriging variance")
k.block <- krige(log(zinc) ~ 1, meuse, meuse.grid, model = m,
                      block=c(400,400))
spplot(k.block["var1.pred"], main = "ordinary kriging predictions")
spplot(k.block["var1.var"],  main = "ordinary kriging variance")
```

I was going to show you some graphics but the lack of a color printer (and photocopy machine) results in gray scale graphs that are not useful. The table below contains predicted values and kriging variances for some selected locations.

Location	*Prediction - Point*	*Variance - Point*	*Prediction - Block*	*Variance - Block*
1	5.923	0.131	6.004	0.016
2	5.765	0.119	5.914	0.023
3	5.640	0.104	6.004	0.023
4	5.616	0.141	5.914	0.034
5	6.877	0.170	5.833	0.046

Note the decrease in the variances with block kriging.