Ordinary Kriging (Ch. 5.5 - Bailey & Gatrell)

The following introduction to the method of ordinary kriging will first present the kriging theory, give a small example, examine the effect of the variogram attributes on kriging, and finally provide a comparison to the other estimation methods for the Walker Lake data.

The prediction methods discussed earlier in the course (polygonal declustering, triangulation, distance-based methods) were all based on weighted averages of some subset of the sampled points. The weights in these cases were based solely on the *locations* of the sampled points. They do not utilize any information about how *similar* the sampled values $y(\boldsymbol{t_i}) = y_i$ are expected to be to $y(\boldsymbol{s_0}) = y_0$, the predicted value. The kriging estimator incorporates the covariance structure among the Y_i's into the weights for predicting Y_0. In this way, the ordinary kriging estimator, like the other estimators studied, is also a weighted average:

$$\widehat{Y}(\boldsymbol{s_0}) = \widehat{Y}_0 = \sum_{i=1}^{n} w_i y_i = \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \boldsymbol{w_1' y},$$

where: y_i = the i^{th} sampled value and w_i = weight for Y_i, $i = 1, \ldots, n$. With kriging, the weights are based on the covariances *among* points in the sample and the covariances *between* sample points and the point to be predicted. To be specific, we need the following:

1. Covariances among points in the sample: $C_{ij} = C(s_i, s_j) = \text{Cov}(Y_i, Y_j)\ \forall i, j$ (n^2 of them)

$$\Leftrightarrow \quad \boldsymbol{C_1} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{12} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n} & C_{2n} & \cdots & C_{nn} \end{bmatrix} = \begin{array}{c} \text{covariance matrix of the } n \\ \text{sample values.} \end{array}$$

   - Note: The covariances C_{ij} are considered to be parameters in this kriging context, which are estimated by a modeled variogram.

2. Covariances between sample points & the prediction point:

$$C_{i0} = \text{Cov}(Y(s_i), Y(s_0)) = C(s_i, s_0) = \text{Cov}(Y_i, Y_0)\ \forall i \ \ (n \text{ of them})$$

$$\Leftrightarrow \quad \boldsymbol{c_0} = \begin{bmatrix} C_{10} \\ C_{20} \\ \vdots \\ C_{n0} \end{bmatrix} = \begin{array}{c} \text{vector of covariances between the} \\ \text{sample points \& the prediction point.} \end{array}$$

Derivation of the Kriging Estimator: The idea behind kriging is to find the estimator $\widehat{Y}_0 = \boldsymbol{w'Y}$ such that:

1. $E(\widehat{Y}_0) = E(Y_0)$ (unbiasedness). This will be satisfied if $\sum w_i = 1$ and the mean is stationary ($E(Y_i) = \mu,\ \forall i$).

2. The prediction variance $\sigma_\epsilon^2 = E[Y_0 - \widehat{Y}_0]^2 = \text{Var}(Y_0 - \widehat{Y}_0)$ is minimized.

- The kriging estimator is the optimal estimator in the sense that it minimizes this prediction or error variance. This quantity is also known as the mean squared prediction error (MSPE).
- Ordinary kriging is an optimal predictor if the mean is assumed constant but unknown over the entire region of interest.
- If the mean is constant but known, a method called simple kriging gives optimal predictions (rarely used in practice).
- Any estimator meeting conditions (1) & (2) above is said to be a best linear unbiased predictor (BLUP).

The mean squared prediction error (MSPE) can be written as:

$$\begin{aligned}
\sigma_\epsilon^2 = \text{Var}(Y_0 - \widehat{Y}_0) &= \text{Var}(Y_0) + \text{Var}(\widehat{Y}_0) - 2\text{Cov}(\widehat{Y}_0, Y_0) \\
&= \sigma^2 + \text{Var}(\textstyle\sum w_i Y_i) - 2\text{Cov}(\sum w_i Y_i, Y_0) \\
&= \sigma^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j \text{Cov}(Y_i, Y_j) - 2\sum_{i=1}^n w_i \text{Cov}(Y_i, Y_0) \\
&= \sigma^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j C_{ij} - 2\sum_{i=1}^n w_i C_{i0}.
\end{aligned}$$

We want to minimize $\text{Var}(Y_0 - \widehat{Y}_0)$ such that $\sum_{i=1}^n w_i = 1$. How? - method of Lagrange multipliers.

1. Form the Lagrangian:
$$\begin{aligned}
\text{L} &= \text{Var}(Y_0 - \widehat{Y}_0) + 2\lambda(\textstyle\sum w_i - 1) \\
&= \sigma^2 + \sum_{i=1}^n \sum_{j=1}^n w_i w_j C_{ij} - 2\sum_{i=1}^n w_i C_{i0} + 2\lambda(\textstyle\sum w_i - 1),
\end{aligned}$$

   where the last term in this sum guarantees that the unbiasedness constraint ($\sum w_i = 1$) is met through the Lagrange multiplier λ.

2. Take partial derivatives of L with respect to the w_i's and λ, set these equal to zero, and solve:

$$\left\{ \begin{array}{c} 2\sum_{j=1}^n w_j C_{1j} - 2C_{10} + 2\lambda = 0 \\ \vdots \quad \vdots \quad \vdots \\ 2\sum_{j=1}^n w_j C_{nj} - 2C_{n0} + 2\lambda = 0 \\ 2\left(\sum_{j=1}^n w_j - 1\right) = 0 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{c} \sum_{j=1}^n w_j C_{1j} + \lambda = C_{10} \\ \vdots \quad \vdots \quad \vdots \\ \sum_{j=1}^n w_j C_{nj} + \lambda = C_{n0} \\ \sum_{j=1}^n w_j \cdot 1 + 0 = 1 \end{array} \right\}$$

$$\Leftrightarrow \left[\begin{array}{ccc|c} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & & \vdots & \vdots \\ C_{1n} & \cdots & C_{nn} & 1 \\ \hline 1 & \cdots & 1 & 0 \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} C_{10} \\ \vdots \\ C_{n0} \\ \hline 1 \end{array}\right] \Leftrightarrow \underbrace{\left[\begin{array}{c|c} \boldsymbol{C_1} & \mathbf{1} \\ \hline \mathbf{1}' & 0 \end{array}\right]}_{\boldsymbol{C}} \underbrace{\left[\begin{array}{c} \boldsymbol{w_1} \\ \hline \lambda \end{array}\right]}_{\boldsymbol{w}} = \underbrace{\left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right]}_{\boldsymbol{D}}.$$

Hence the weights for ordinary kriging can be found from the following system of linear equations, known as the kriging equations:

$$\boldsymbol{w} = \left[\frac{\boldsymbol{w_1}}{\lambda}\right] = \left[\begin{array}{c} w_1 \\ \vdots \\ \hline w_n \\ \hline \lambda \end{array}\right] = \underbrace{\left[\begin{array}{ccc|c} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & & \vdots & \vdots \\ C_{n1} & \cdots & C_{nn} & 1 \\ \hline 1 & \cdots & 1 & 0 \end{array}\right]^{-1}}_{\boldsymbol{C}^{-1}} \underbrace{\left[\begin{array}{c} C_{10} \\ \vdots \\ \hline C_{n0} \\ \hline 1 \end{array}\right]}_{\boldsymbol{D}} = \left[\begin{array}{c|c} \boldsymbol{C_1} & \boldsymbol{1} \\ \hline \boldsymbol{1'} & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right]$$

where $C_{ij} = \text{Cov}(Y_i, Y_j)$, $C_{i0} = \text{Cov}(Y_i, Y_0)$, and λ is a Lagrange multiplier that appears due to the unbiasedness constraint $\sum w_i = 1$.

Kriging Estimator of the Global Mean: Suppose the response variables $\boldsymbol{y}' = (Y_1, \ldots, Y_n)$ are multivariate normally distributed with a common mean given by: μ. Since this common mean represents the *global* mean of the region of study, we would like to estimate it.

- In a regression setting, we generally have a set of n independent observations from which to estimate the global mean μ. With spatial data under the kriging assumption of a stationary mean, it turns out that the generalized least squares estimate, given by:

  $$\widehat{m}_{GLS} = (\boldsymbol{1'C}_1^{-1}\boldsymbol{1})^{-1}\boldsymbol{1'C}_1^{-1}\boldsymbol{y},$$

  is the kriging estimate of the mean for the entire region.

- It should be noted that the kriging estimator is both globally and conditionally *unbiased* whenever the data are normally distributed.

Prediction Variance and Confidence Intervals: We obtained the kriging equations by minimizing the variance of the prediction errors, given by:

$$\sigma_\epsilon^2 = \text{Var}(Y_0 - \widehat{Y}_0) = \sigma^2 + \sum_i \sum_j w_i w_j C_{ij} - 2\sum_i w_i C_{i0}$$

subject to the constraint that $\sum w_i = 1$. Having found a solution to these equations, it would be of interest to know what this minimum prediction variance is. To find this prediction variance corresponding to the kriging estimates, recall the form of the kriging equations, given below:

$$\left\{\begin{array}{ccccc} \sum_j w_j C_{1j} & + & \lambda & = & C_{10} \\ \vdots & & \vdots & & \vdots \\ \sum_j w_j C_{nj} & + & \lambda & = & C_{n0} \\ \sum_j w_j & + & 0 & = & 1 \end{array}\right\}$$

Notice that if we multiply equation i by w_i and sum the first n equations, we have:

$$\sum_i w_i \sum_j w_j C_{ij} + \sum_i w_i \lambda = \sum_i w_i C_{i0} \implies \sum_i \sum_j w_i w_j C_{ij} = \sum_i w_i C_{i0} - \lambda.$$

Substituting this into the equation above for the prediction variance, we have:

$$\begin{aligned}\sigma^2_\epsilon &= \sigma^2 + \sum_i \sum_j w_i w_j C_{ij} - 2\sum_i w_i C_{i0} = \sigma^2 - \left(\sum_{i=1}^n w_i C_{i0} + \lambda\right) \\ &= \sigma^2 - \left[\begin{array}{ccc|c} w_1 & \cdots & w_n & \lambda \end{array}\right] \left[\begin{array}{c} C_{10} \\ \vdots \\ C_{n0} \\ \hline 1 \end{array}\right] = \sigma^2 - \left[\begin{array}{c|c} \boldsymbol{w_1} & \lambda \end{array}\right] \left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right] = \sigma^2 - \boldsymbol{w'D}.\end{aligned}$$

- This prediction variance goes by several names: the prediction variance, the kriging variance, and the mean squared prediction error, among the more common ones.
- If the response variables are multivariate normally distributed, approximate $100(1-\alpha)\%$ prediction intervals can be constructed via: $\widehat{v}_0 \pm z_{1-\alpha/2} \cdot \sigma_R$, where $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)$ quantile of the standard normal distribution.
- It should be emphasized that these prediction intervals are approximate whenever σ^2_ϵ is estimated rather than assumed known, and/or when the joint distribution of the Y_i's is not normal. In practice, we will need to estimate σ^2_ϵ and the responses will often not be normal, so these intervals are really only appropriate for large sample sizes.

Writing the Kriging Equations in Terms of the Semivariograms: Since we modeled the correlation structure in the data in terms of a semivariogram, not a covariogram, we would like to be able to write these kriging equations in terms of the semivariances instead of the covariances. Under second-order stationarity, we showed earlier that the semivariogram may be written in terms of the covariance function as:

$$\begin{aligned}\gamma_{ij} &= \frac{1}{2}\text{Var}(V_i - V_j) = \frac{1}{2}\left(\sigma^2 + \sigma^2 - 2C_{ij}\right) \\ &= \sigma^2 - C_{ij} = C_{ii} - C_{ij} \\ &\Rightarrow \underline{C_{ij} = \sigma^2 - \gamma_{ij}}.\end{aligned}$$

Instead of actually converting the semivariances to covariances, it is more useful to rewrite the kriging equations in terms of the semivariogram. The equations turn out to have a similar form as for the covariances. To see how this works, note the following:

$$\boldsymbol{C_1} = \left[\begin{array}{ccc} C_{11} & \cdots & C_{1n} \\ \vdots & & \vdots \\ C_{n1} & \cdots & C_{nn} \end{array}\right] = \left[\begin{array}{ccc} \sigma^2 & \cdots & \sigma^2 \\ \vdots & & \vdots \\ \sigma^2 & \cdots & \sigma^2 \end{array}\right] - \left[\begin{array}{ccc} \gamma_{11} & \cdots & \gamma_{1n} \\ \vdots & & \vdots \\ \gamma_{n1} & \cdots & \gamma_{nn} \end{array}\right] = \sigma^2\mathbf{1}\mathbf{1}' - \boldsymbol{\Gamma_1}, \text{ and :}$$

$$\boldsymbol{c} = \left[\begin{array}{c} C_{10} \\ \vdots \\ C_{n0} \end{array}\right] = \left[\begin{array}{c} \sigma^2 \\ \vdots \\ \sigma^2 \end{array}\right] - \left[\begin{array}{c} \gamma_{10} \\ \vdots \\ \gamma_{n0} \end{array}\right] = \sigma^2\mathbf{1} - \boldsymbol{\Gamma}_0, \text{ where: } \mathbf{1'} = (1, 1, \ldots, 1).$$

Using these two matrix relationships, the kriging equations can be rewritten in terms of the

semivariances as:

$$\left[\begin{array}{c|c} \boldsymbol{C_1} & \mathbf{1} \\ \hline \mathbf{1'} & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{w_1} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right] \implies \left[\begin{array}{c} \boldsymbol{C_1 w_1} + \lambda \mathbf{1} \\ \hline \mathbf{1'w_1} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right]$$

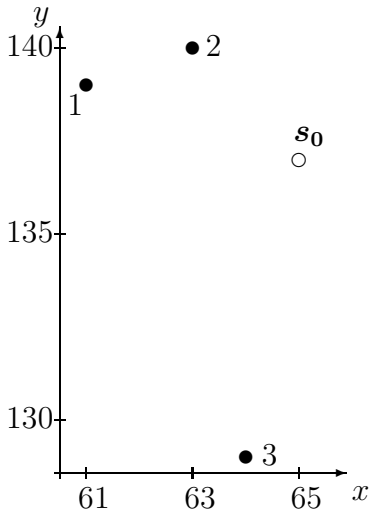
$$\implies \left[\begin{array}{c} \sigma^2 \mathbf{1}\mathbf{1'}\boldsymbol{w_1} - \boldsymbol{\Gamma_1 w_1} + \lambda \mathbf{1} \\ \hline \mathbf{1'w_1} \end{array}\right] = \left[\begin{array}{c} \sigma^2 \mathbf{1} - \boldsymbol{\Gamma}_0 \\ \hline 1 \end{array}\right]$$

$$\implies \left[\begin{array}{c} -\boldsymbol{\Gamma_1 w_1} + \lambda \mathbf{1} \\ \hline \mathbf{1'w_1} \end{array}\right] = \left[\begin{array}{c} -\boldsymbol{\Gamma}_0 \\ \hline 1 \end{array}\right]$$

$$\implies \left[\begin{array}{c|c} -\boldsymbol{\Gamma_1} & \mathbf{1} \\ \hline \mathbf{1'} & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{w_1} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c} -\boldsymbol{\Gamma}_0 \\ \hline 1 \end{array}\right].$$

- It should be noted that these two forms of the kriging equations are completely equivalent under second-order stationarity. It is simply a matter of whether we work with the covariances or the semivariograms.

A Small Example: In [I & S], an example of how ordinary kriging works is given for a data set of 7 sample values. As an even smaller example to illustrate the mechanics of ordinary kriging, suppose we consider just a subset of 3 of these 7 points, as given in the table below, and want to predict the value of V at the new location $\boldsymbol{s_0}$ = (65E, 137N).



Sample	x	y	V	Distance to $\boldsymbol{s_0}$
1	61	139	477	4.47
2	63	140	696	3.61
3	64	129	227	8.06

To compute kriging estimates, we will need the covariances among all points and between each of the observed points and the point to be predicted. The usual way to obtain these is through a covariance function, such as the commonly used exponential covariance function. Recall that the exponential semivariogram had the form:

$$\gamma(\boldsymbol{h}) = \left\{\begin{array}{ll} 0 & \text{if } |\boldsymbol{h}| = 0 \\ a + (\sigma^2 - a)\left(1 - \exp\left(\frac{-3|\boldsymbol{h}|}{r}\right)\right) & \text{if } \boldsymbol{h} > 0 \end{array}\right\}, \quad \text{where:}$$

a = nugget effect, r = range, and σ^2 = sill or variance where there is no correlation present. The corresponding exponential covariance function has the form:

$$C(\boldsymbol{h}) = \left\{\begin{array}{ll} a + (\sigma^2 - a) & \text{if } |\boldsymbol{h}| = 0 \\ (\sigma^2 - a)\exp\left(\frac{-3|\boldsymbol{h}|}{r}\right) & \text{if } \boldsymbol{h} > 0 \end{array}\right\}.$$

What does this function look like?

Suppose for this small example that the nugget effect a is 0, the range r is 10, and the sill $\sigma^2 = 100$. Then this exponential covariance function becomes:

$$C(\boldsymbol{h}) = 100e^{-0.3|\boldsymbol{h}|}, \;\; |\boldsymbol{h}| \geq 0.$$

Computing with this covariance function then, the covariance matrix among the three observed points is:

$$\boldsymbol{C}_1 = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{bmatrix} 100 & 51.1 & 4.4 \\ & 100 & 3.6 \\ & & 100 \end{bmatrix}.$$

- In general, $C_{ii} = \text{Cov}(V_i, V_i) = \text{Var}(V_i) = C(0) = 100$, for all i.
- The points in the lower triangular portion of this matrix are omitted due to the symmetry of the covariance matrix.

The covariances between the three sample points and $V_0 = V(\boldsymbol{s}_0)$ are:

$$\boldsymbol{c} = \begin{bmatrix} C_{10} \\ C_{20} \\ C_{30} \end{bmatrix} = \begin{bmatrix} 26.1 \\ 33.9 \\ 8.00 \end{bmatrix}.$$

The estimate of the global mean is found as follows:

$$\widehat{m}_{GLS} = \frac{\mathbf{1}'\boldsymbol{C_1}^{-1}\boldsymbol{v}}{\mathbf{1}'\boldsymbol{C_1}^{-1}\mathbf{1}} = \underline{434},$$

where $\boldsymbol{v}$ is the vector of observed values: $\boldsymbol{v}' = [477, 696, 227]$. As pointed out earlier, this is nothing more than the generalized least squares estimator of the mean of V.

- Note: The inverse of a matrix can be found in **R** using the **solve** function as indicated on the **R** handout given earlier in the semester (e.g.: typing **Cinv** <− **solve(C)** will compute the inverse of **C**).
- Question: What is the sample mean of $\boldsymbol{v}$? Why is there a difference between this and the GLS estimator found above, and which do you prefer?

Deriving the Kriging Weights: Recall that in matrix form, the kriging equations can be written as:

$$\left[\begin{array}{c} \boldsymbol{w_1} \\ \hline \lambda \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{C_1} & \mathbf{1} \\ \hline \mathbf{1}' & 0 \end{array}\right]^{-1} \left[\begin{array}{c} \boldsymbol{c_0} \\ \hline 1 \end{array}\right],$$

where after inverting the partitioned matrix $\left[\begin{array}{c|c} \boldsymbol{C_1} & \mathbf{1} \\ \hline \mathbf{1}' & 0 \end{array}\right]$ (as assigned for homework for the math graduate students), the weights can be written as:

$$\boldsymbol{w_1} = \boldsymbol{C_1}^{-1}\boldsymbol{c_0} - \frac{\boldsymbol{C_1}^{-1}\mathbf{1}\mathbf{1}'\boldsymbol{C_1}^{-1}\boldsymbol{c_0}}{\mathbf{1}'\boldsymbol{C_1}^{-1}\mathbf{1}} + \frac{\boldsymbol{C}^{-1}\mathbf{1}}{\mathbf{1}'\boldsymbol{C_1}^{-1}\mathbf{1}}.$$

Multiplying all of this out via **R** for our 3-point example with an exponential covariance function, the kriging weights were found to be:

$$\boldsymbol{w_1} = \begin{bmatrix} 0.2677 \\ 0.4309 \\ 0.3014 \end{bmatrix}. \quad \text{Note that the weights sum to 1.}$$

These particular weights do not seem very interesting until they are compared with the strictly inverse distance-based weights as given in the table below.

	Weights (w_i)			
Sample Point	Inverse Distance	Kriging	Distance	Location
1	.36	.27	4.5	(61,139)
2	.45	.43	3.6	(63,140)
3	.20	.30	8.1	(64,129)

- Note that point 1 is much closer to $\boldsymbol{s}_0$ than point 3, yet has a smaller kriging weight. Can you explain this apparent reversal in the ranks of the weights with ordinary kriging?

The predicted value of V at $\boldsymbol{s}_0$ is:

$$\begin{aligned} \widehat{v}_0 &= \sum_{j=1}^{n} w_j v_j = \boldsymbol{w_1}'\boldsymbol{v} = \boldsymbol{c_0}'\boldsymbol{C_1}^{-1}\boldsymbol{v} - \frac{\boldsymbol{c_0}'\boldsymbol{C_1}^{-1}\boldsymbol{1}\boldsymbol{1}'\boldsymbol{C_1}^{-1}\boldsymbol{v}}{\boldsymbol{1}'\boldsymbol{C_1}^{-1}\boldsymbol{1}} + \frac{\boldsymbol{1}'\boldsymbol{C_1}^{-1}\boldsymbol{v}}{\boldsymbol{1}'\boldsymbol{C_1}^{-1}\boldsymbol{1}} \\ &= \widehat{m}_{GLS} + \boldsymbol{c_0}'\boldsymbol{C_1}^{-1}(\boldsymbol{v} - \widehat{m}_{GLS}\boldsymbol{1}). \end{aligned}$$

- This says that the best predictor of V at a specific site $\boldsymbol{s}_0$, is the estimated global mean, $\widehat{m}_{GLS}$, plus an *adjustment* due to correlations with and among the nearby sites. For this 3-point example, the first equation above can be used to compute:

$$\widehat{v}_0 = \boldsymbol{w_1}'\boldsymbol{v} = (.2677)(477) + (.4309)(696) + (.3014)(227) = \underline{496}.$$

The prediction variance is:

$$\begin{aligned} \sigma^2_\epsilon &= \text{Var}(V(\boldsymbol{s}_0) - \widehat{V}(\boldsymbol{s}_0)) = \sigma^2 - \begin{bmatrix} \boldsymbol{w_1}' & \lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{c_0} \\ 1 \end{bmatrix} \\ &= 100 - \begin{bmatrix} .2677 & .4309 & .3014 & -2.3972 \end{bmatrix} \begin{bmatrix} 26.1 \\ 33.9 \\ 8.00 \\ 1 \end{bmatrix} = \underline{78.3915}. \end{aligned}$$

This gives a 95% prediction interval of:

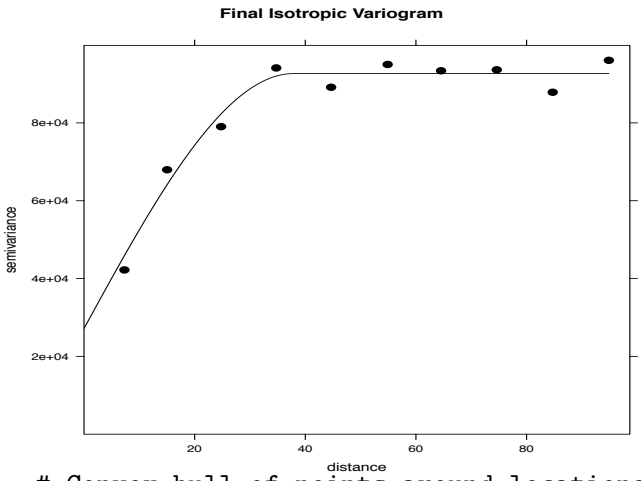
$$\widehat{v}_0 \pm z_{.975}\sqrt{\sigma_\epsilon^2} \implies 496 \pm (1.96)\sqrt{78.3915} = \boxed{(479,\ 513)}.$$

- The only elements that entered into the variance σ_ϵ^2 are the variances of V (σ^2), and the hypothesized covariance function. The *values* v_i only come into play in the estimation of σ^2 and $C(\boldsymbol{h})$.
- The data locations have a major impact on the prediction variance through $C(\boldsymbol{h})$. The assumed form of the covariance function also has a major effect on the prediction variances.
- The fact that the data values v_i have no direct effect on the prediction variance is an artifact of using an estimator which assumes that $C(\boldsymbol{h})$ is a known function.

Kriging in **R**: To illustrate how ordinary kriging is performed in **R**, consider the Walker Lake V-concentration sample data. In an earlier handout, the final isotropic semivariogram was found to be of approximately spherical form with geometric anisotropy present. The range ratio corresponding to the distances of maximum and minimum spatial continuity was roughly 2.4, and the data were rotated clockwise by 14 degrees to align these directions at 0 and 90 degree azimuths. Reconstructing this geometric anisotropy-corrected variogram but without the rotation, the variogram was fit using the following sequence of commands:

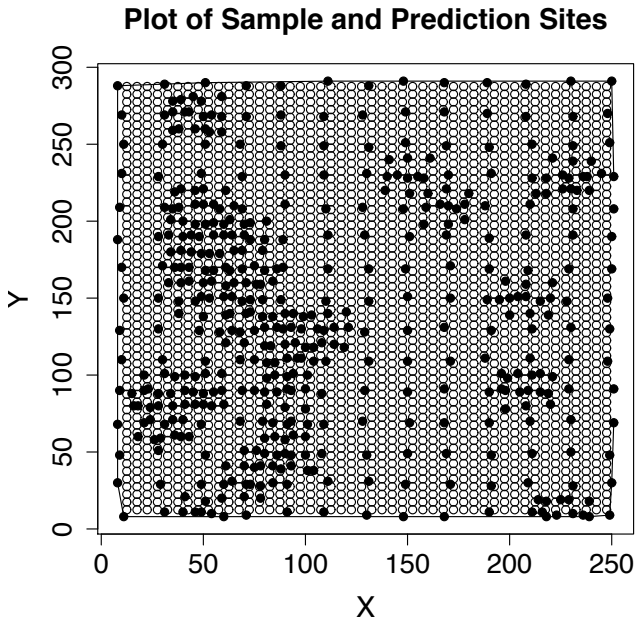
```
coordinates(walk470) = ~x+y                # Assigns coordinates for use in "variogram".
walk.var2 <- variogram(v ~ x+y,            # Computes omnidirectional variogram of V
  data=walk470,width=10,cutoff=100)        #   with lags and maxdist specified.
v <- walk470$v; u <- walk470$u             # Gives the Walker Lake values short names.
xy <- coordinates(walk470)                 # Defines an nx2 vector of (x,y) loc's.
az <- seq(0,165,15)                        # Defines azimuths from 0 to 165 by 15.
walk.var4 <- variogram(v~newx+newy,        # Computes the variogram for the V data
  data=walk470, width=10, alpha=az,        #   width lag 10, maximum distance 100,
  tol.hor=15, cutoff=100)                  #   angles in "az" an angle tolerance 15.
ranges <- rose(walk.var4,80000)            # Generates a rose diagram for g=80000.
xyplot(gamma ~ dist|as.factor(dir.hor),    # Plots directional variograms with loess
  data=walk.var4,layout=c(4,3), panel =    #   fits through the data and prints
  panel.gamma0, gamma0=80000,              #   interpolated range at gamma specified.
  main="Interpolated Directional Ranges")
ratio <- 2.9                               # Range ratio based on rose diagram.
model1.out <- fit.variogram(walk.var4,     # Fits an anisotropic variogram model with
  vgm(70000,"Sph",40,20000,anis=           #   major axis at 166-degrees, and range
  c(166,1/ratio)),fit.method=2)            #   ratio given by "ratio".
plot(walk.var4,model1.out,main=            # Plots the directional variograms with
  "Anisotropy-Corrected Variograms")       #   fitted corrected variogram models.
plot(walk.var2,model1.out,cex=1.3,main=    # Plots the final isotropic variogram
  "Final Isotropic Variogram",pch=16)      #   using anisotropy-corrected distances.
```

As a reminder of the quality of fit of this variogram model, the final spherical model used is plotted to the right. With this final variogram in place describing the covariance structure of the data, kriging can be performed in two steps. For these Walker Lake rotated V-concentrations, ordinary kriging is performed by issuing the commands:




```
poly <- chull(coordinates(walk470))          # Convex hull of points around locations.
plot(coordinates(walk470),type="n",          # Creates a blank (type="n") plot of
  xlab="X",ylab="Y",cex.lab=1.6,main=        #   coordinates with axis labels and a
  "Plot of Sample and Prediction Sites",     #   title.
  cex.axis=1.5,cex.main=1.6)
lines(coordinates(walk470)[poly,])           # Plots the convex hull boundary.
poly.in <- polygrid(seq(2.5,247.5,5),        # Creates a grid of points inside the
  seq(2.5,297.5,5),                          #   "poly" polygon according to the limits
  coordinates(walk470)[poly,])               #   given in the "seq" functions.
points(poly.in)                              # Plots the prediction points.
points(coordinates(walk470),pch=16)          # Plots the sample points (closed circles).
coordinates(poly.in) <- ~ x+y                # Converts poly.in to coordinates object.
krige.out <- krige(v ~ 1, walk470,           # Kriges the Walker Lake data at the points
  poly.in, model=model1.out)                 #   given in "poly.in", returning predictions
                                             #   and kriging variances.
krige.out                                    # Prints the kriging output.
```

```
> krige.out
        coordinates  var1.pred var1.var
1      (12.5, 12.5)  101.59957 64341.79
2      (17.5, 12.5)  190.48362 85151.15
3      (22.5, 12.5)  232.52599 88003.04
 .           .            .        .
158    (77.5, 27.5)  676.13827 43843.91
159    (82.5, 27.5)  598.32125 53236.96
160    (87.5, 27.5)  378.91797 53917.84
161    (92.5, 27.5)  240.26011 52615.27
162    (97.5, 27.5)  383.65440 68910.91
163   (102.5, 27.5)  486.18065 64224.79
 .           .            .        .
2686 (237.5, 287.5)  174.38720 82686.29
2687 (242.5, 287.5)  166.54622 81208.45
2688 (247.5, 287.5)  131.04461 67179.65
```



To make predictions at specific sites: Suppose we choose the three locations (60,190), (225,50), (110,185) at which to make kriging predictions. These sites are shown as darkened circles on the greyscale image on the next page and were included as they represent areas of high, medium, and low values in the data. To krige these three points, the following code can be used.

```
predpts <- matrix(c(60,190,225,50,110,    # Creates a 3x2 matrix of prediction locations:
  185),ncol=2,byrow=T)                    #   (60,190), (225,90), (110,185).
predpts.g <- data.frame(x=predpts[,1],    # Creates a data frame of the 3 (x,y)
  y=predpts[,2])                          #   locations.
coordinates(predpts.g) <- ~x+y            # Defines "predpts.g" as a spatial object.
g <- gstat(NULL,"new.v",v~1,data=walk470, # Creates a geostats object incorporating
  model=model1.out)                       #   the variogram info.
three.pred <- predict(g,predpts.g)        # Makes kriging predictions at the 3 points.
three.pred <- data.frame(three.pred,se=   # Adds the kriging standard errors to the
  sqrt(three.pred[,4]))                   #   data frame.
three.pred                                # Prints kriging predictions.
> three.pred
    x   y new.v.pred new.v.var       se
1  60 190  1284.9958  42531.35 206.2313
2 225  50   308.1408  75610.49 274.9736
3 110 185   108.7244  64114.22 253.2079
```

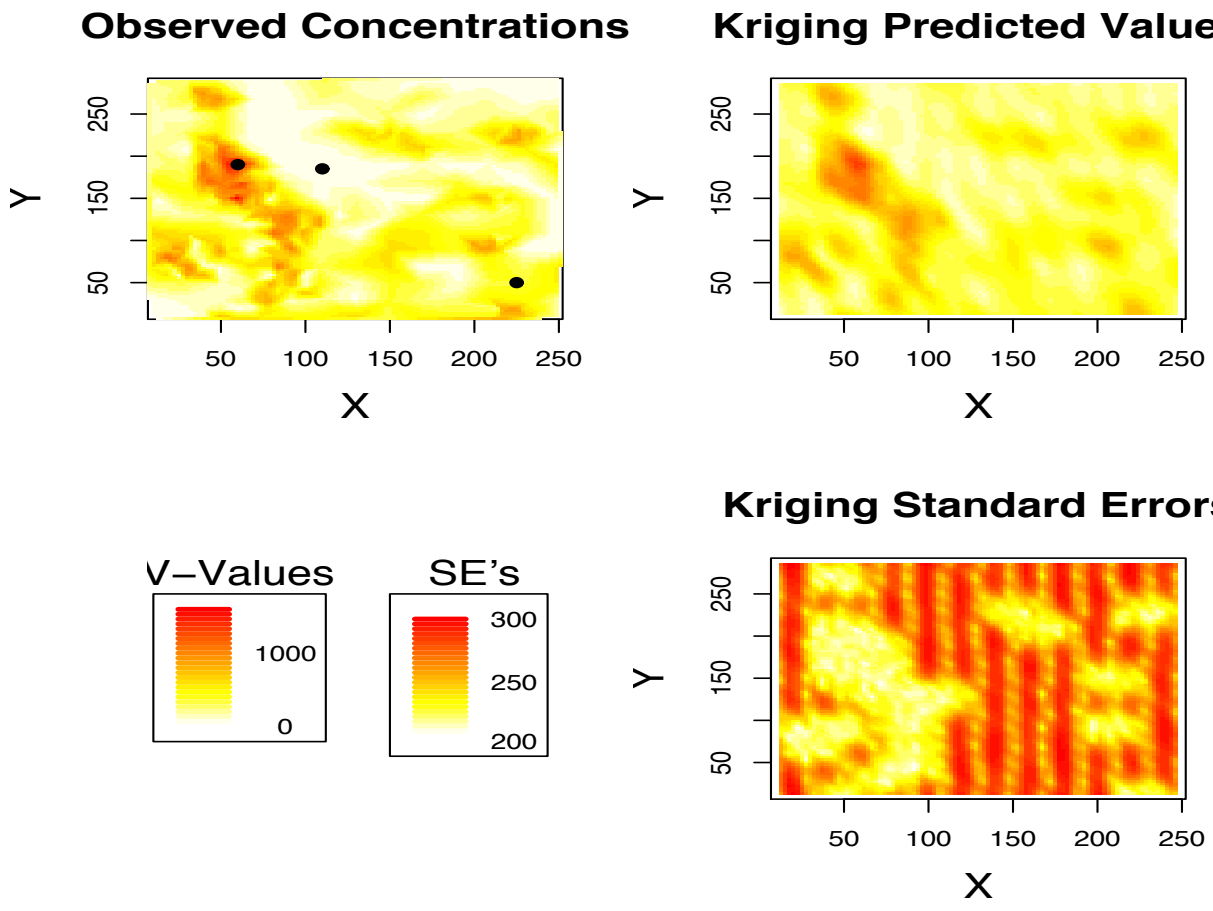
- Do these standard errors seem large? Does this make sense?

- Why is the standard error smallest for the largest V-value?

To create a kriging surface and corresponding standard errors: The following code was used to generate three image maps of the original V-values, kriging predictions, and kriging standard errors. The plots follow.

```
xpred <- seq(min(x),max(x),length=100)    # X-coordinates for interpolation locations.
ypred <- seq(min(y),max(y),length=100)    # Y-coordinates for interpolation locations.
int.obs <- interp(x,y,v,xo=xpred,yo=ypred)# Interpolates the observed V-values.
int.pred <- interp(coordinates(poly.in)[, # Interpolates the kriging prediction values
  1],coordinates(poly.in)[,2],krige.out$  #   over the coordinates in xo and yo.
  var1.pred,xo=xpred,yo=ypred)
int.se <- interp(coordinates(poly.in)[,1],# Interpolates the kriging standard errors
  coordinates(poly.in)[,2],sqrt(krige.out$#   over the coordinates in xo and yo.
  var1.var),xo=xpred,yo=ypred)
par(mfrow=c(2,2))                         # Creates a 1x3 graphics window.
zmin <- min(int.obs$z[!is.na(int.obs$z)], # Computes the minimum V/predicted value,
```

```
  int.pred$z[!is.na(int.pred$z)])          #   avoiding missing values (!is.na).
zmax <- max(int.obs$z[!is.na(int.obs$z)], # Computes the maximum V/predicted value,
  int.pred$z[!is.na(int.pred$z)])          #   avoiding missing values (!is.na).
image(int.obs,xlab="X",ylab="Y",cex.lab=1.6,main=   # Creates a greyscale plot of the
  "Observed Concentrations",cex.main=1.6,           #   interpolated V-values with axis
  zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1) #   labels and a title.
points(predpts[,1],predpts[,2],pch=16)              # Overlays three prediction sites.
image(int.pred,xlab="X",ylab="Y",cex.lab=1.6,main=  # Creates a greyscale plot of the
  "Kriging Predicted Values",cex.main=1.6,          #   interpolated kriged V-values
  zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1) #   with axis labels and a title.
image(int.se,xlab="",ylab="",axes=F,zlim=range(     # Creates a blank image plot
  krige.out$var1.var),col=rev(heat.colors(24)))     #   via "axes=F".
text(60,275,"V-Values",cex=1.6)                     # Puts title on the V-legend.
text(200,275,"SE's",cex=1.6)                        # Puts title on the SE-legend.
image.legend(10,250,zlim=c(zmin,zmax),              # Plots a legend at (10,250) for
  col=rev(heat.colors(24)))                         #   the V-values.
image.legend(150,250,zlim=sqrt(range(               # Plots a legend at (150,250) for
  krige.out$var1.var)),col=rev(heat.colors(24)))    #   the standard errors.
image(int.se,xlab="X",ylab="Y",cex.lab=1.6,main=    # Creates a greyscale plot of the
  "Kriging Standard Errors",cex.main=1.6,           #   interpolated kriging SE's
  col=rev(heat.colors(24)),cex=1)                   #   with axis labels and a title.
```



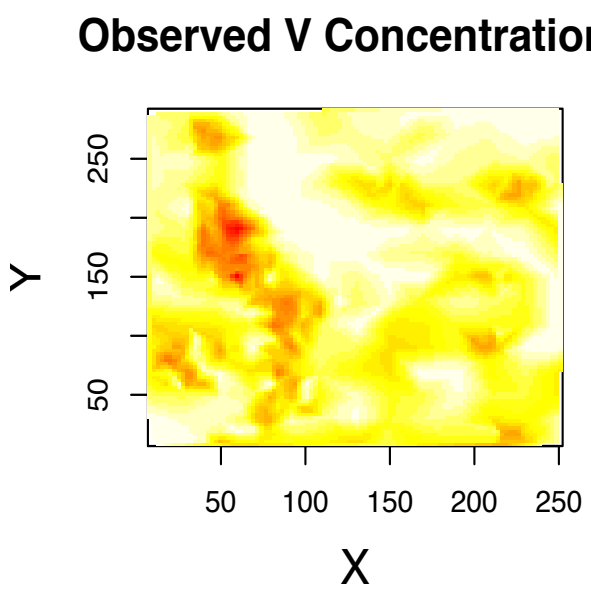
- How do the kriging predictions look? What problems do you see? Why is kriging unable to capture the interpolated spatial pattern?

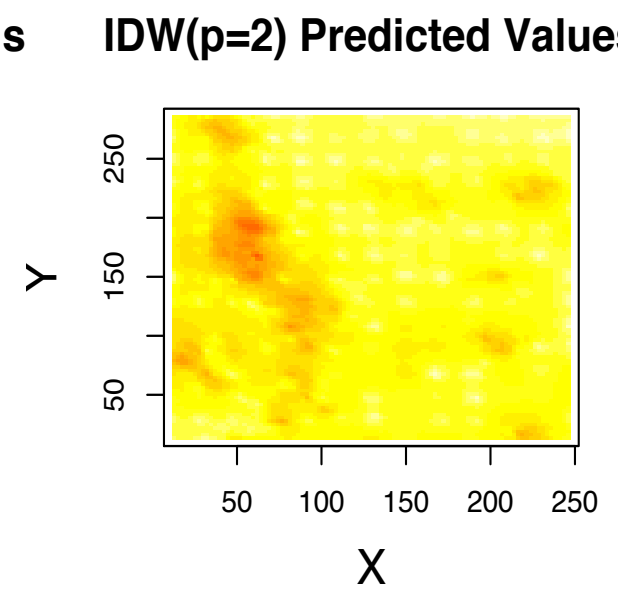
- What patterns do you see in the kriging standard errors? Does this make sense?

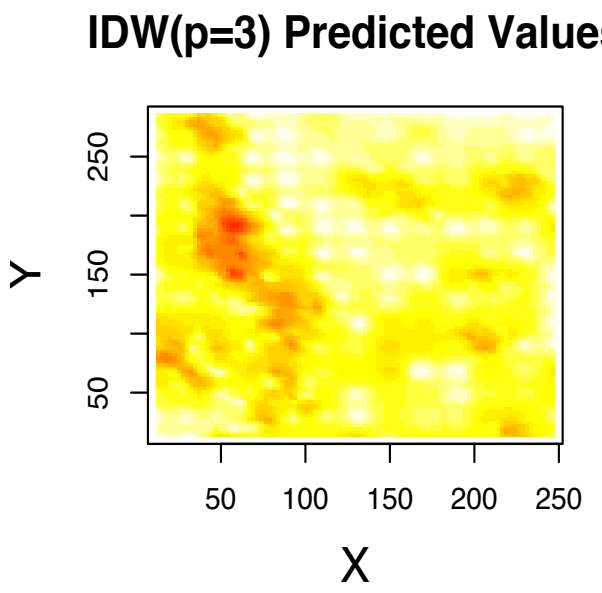
How does ordinary kriging compare to inverse distance weighting? The **idw** function can be used to perform inverse distance weighting in much the same way as the **krige** function, with one exception. Instead of requiring a variogram model, it requires the power to be used in the inverse distance weighting. Code to do this is given below, followed by plots of the inverse distance weighted predictions for three different powers on the next page.

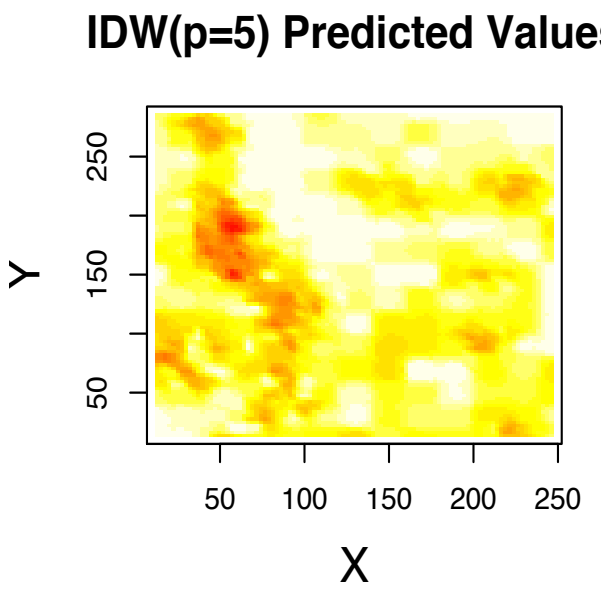
```
idw.out1 <- idw(v ~ 1,walk470,poly.in,idp=2)  # Inverse dist. wt. predictions (p=2)
idw.out2 <- idw(v ~ 1,walk470,poly.in,idp=3)  # Inverse dist. wt. predictions (p=3)
idw.out3 <- idw(v ~ 1,walk470,poly.in,idp=5)  # Inverse dist. wt. predictions (p=5)
int.pred1 <- interp(coordinates(poly.in)[,1],coordinates(poly.in)[,2],
  idw.out1$var1.pred,xo=xpred,yo=ypred)
int.pred2 <- interp(coordinates(poly.in)[,1],coordinates(poly.in)[,2],
  idw.out2$var1.pred,xo=xpred,yo=ypred)
int.pred3 <- interp(coordinates(poly.in)[,1],coordinates(poly.in)[,2],
  idw.out3$var1.pred,xo=xpred,yo=ypred)
par(mfrow=c(2,2))
image(int.obs,xlab="X",ylab="Y",cex.lab=1.6,main="Observed V Concentrations",
  cex.main=1.5,zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1)
image(int.pred1,xlab="X",ylab="Y",cex.lab=1.6,main="IDW(p=2) Predicted Values",
  cex.main=1.5,zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1)
image(int.pred2,xlab="X",ylab="Y",cex.lab=1.6,main="IDW(p=3) Predicted Values",
  cex.main=1.5,zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1)
image(int.pred3,xlab="X",ylab="Y",cex.lab=1.6,main="IDW(p=5) Predicted Values",
  cex.main=1.5,zlim=c(zmin,zmax),col=rev(heat.colors(24)),cex=1)
```

```
> idw.out1[1:3,]
   coordinates var1.pred var1.var
1 (12.5, 12.5)  181.4364       NA
2 (17.5, 12.5)  264.4042       NA
3 (22.5, 12.5)  278.3362       NA
```









- Why do the maps with inverse distance weighting seem to improve as we increase the power?