

ASSIGNMENT 5

(Due on Tuesday, April 12)

This assignment involves only a single (but somewhat more detailed) study that is intended to synthesize most of what you have learned about continuous spatial data analysis. The ultimate objective of this study is to carry out a **geostatistical kriging** of the England and Wales Temperature data from the month of August, 1981, discussed in B&G (p.147,201). [A similar data set is studied by Upton and Fingleton (Item 18 in the References on the Class Web Page, pp. 325-331).] For additional information, try Googling topics like “England Climate”, etc.

The only data available here is the coordinate data (X,Y) for each measurement location. A more suitable geostatistical kriging model would of course involve other relevant explanatory variables (altitude, etc.). So the present model should be considered simply for demonstration purposes in your pedagogical study. The data can be viewed in ARCMAP by opening the file **..projects/eng_temp/eng_temp.mxd**.

- (1) As in previous assignments, make a subdirectory in your home directory, **S:/home/eng_temp**, and then copy all files in **..projects/eng_temp** of the form (**eng_temp_81.***, **mask.***, **eng_temp_bnd.***). Use these to make a map document similar to **eng_temp.mxd**. The order of the layers from top to bottom should be as above from left to right. Unlike the South American data in Assignment 4, the map units here are unknown.¹ For convenience you may wish to reset the displayed map units to “Kilometers” [using the same procedure as in part (b).3.a of Problem 2 in Assignment 4.]. But in fact 100 *km* equals about 80 of these unknown map units.
- (2) To analyze this data, it is convenient to begin with a standard **OLS** analysis in JMP. Start by opening the file, **Eng_temp.jmp**, in the class directory, **..sys502/jmpin**.
 - (a) First check the normality of the **81 Temp** data with **Analyze → Distribution** and then apply the **Normal Quantile Plot** option to the **81 Temp** data (as done in Assignment 4). Comment on the relevance of your findings.
 - (b) Next you will try an exploratory regression analysis to fit this data with a *second-order (quadratic) trend function* of the coordinates. To do so, use **Fit Model** with dependent variable, **81 Temp**, and explanatory variables, (**X, Y, X², XY, Y²**).
 - (c) Notice that both **X** and **X²** are significant, but that *no* variables involving **Y** are significant. To check for possible collinearity effects here, regress **Y²** on **Y** and take a look at both the *p-value* on **Y** and the *R-squared value* for this regression. Recalling that in simple regression the *sample correlation* between the dependent and explanatory variables is precisely the square root of the R-squared value, comment on the possibility of multicollinearity here.

¹ The original map was digitalized by Bailey and Gatrell and presented with no map unit specified.

- (d) Now try eliminating Y^2 from the original regression. What happens to the *p-value* on Y , and to the *adjusted R-Squared* value for the regression? Comment on these effects.
- (e) Finally, eliminate XY from the regression as well, and again comment on both the adjusted R-square effects and the p-value effects for Y . Save the residuals of this regression as a new column, **Res**, for later use.
- (f) In view of these findings, our final quadratic trend model for **81 Temp** will involve only the explanatory variables (X, Y, X^2).

NOTE: Before proceeding, it is important to emphasize that while our subsequent analysis will attempt to refine the regression results above, OLS is almost always the best way to *begin* such an analysis. In particular, OLS generally gives a sufficiently good picture of the *relative* significance among candidate variables to allow the specification of reasonably good models (as was done above).

- (3) Next we shall analyze the residuals from the regression above to gauge the degree of spatial dependency left to be accounted for.
 - (a) To begin, it is appropriate to carry out the usual *nearest-neighbor residual analysis*, with which you are by now very familiar. To do so, observe that the nearest neighbors of each temperature site have been calculated in MATLAB (using the program **neighbors.m** with L = temperature locations and $k = 1$), and have been included as the column, **n_neigh**, in **Eng_temp.jmp**. Using **n_neigh** together with **Res** above, make a new column, **nn_Res**, as in previous assignments, regress **Res** on **nn_Res**, and discuss the relevance your findings. As you should see, these results suggest that **OLS** is actually performing very well in the present case.
 - (b) However, while this simple test is often very useful, we are now in a position to examine spatial dependencies somewhat more carefully. To do so, we begin by fitting a *spherical variogram* to these OLS residuals in MATLAB.
- (4) First open MATLAB and import the text file **eng_temp_81.txt** (which can be found in the directory **..sys502/matlab**). As usual, use **File** → **Import Data**, and check the workspace to be sure that the matrix **eng_temp_81** appears. While the residuals from the **OLS** regression in JMP could be imported to MATLAB for analysis, it is instructive to repeat this basic regression in MATLAB.
 - (a) If you look at the first few rows of this matrix [**» eng_temp_81(1:3,:)**] you will see that the coordinates $[X, Y]$ are the first two columns, and the temperature data is the last column. So to construct the matrix of *explanatory variables*, write:

» x = eng_temp_81(:,1);

```
» y = eng_temp_81(:,2);  
» X0 = [x,y,x.^2];
```

The third command defines the matrix of variables (X, Y, X^2) at the measurement points. Note in particular that the period in the term $x.^2$ specifies component-wise operations, so that each element of the vector x is squared.

(b) The set of *locations*, $L0$, is then given simply by

```
» L0 = [x,y];
```

(c) Finally, define the *temperature values*, $y0$, at the measurement points by

```
» y0 = eng_temp_81(:,3);
```

(d) To perform a multiple regression in MATLAB, you will use the program, **ols_ts.m** (where “**ts**” here denotes my own modification of the **ols.m** program in MATLAB). Open this program in MATLAB either by using **File** → **Open** → **ols_ts**, or by typing

```
>> edit ols_ts
```

Here you will see that the inputs include an information structure, **info**, providing a number of useful options. Here we will only use the first option, namely to construct names for all variables. To do so, write:

```
>> info.names = strvcat('Temp','X','Y','X^2');
```

Notice also that the output of this regression program is a structure, **results**, which contains useful data results. In particular, the fifth line of the output description shows that **results.resids** contains the desired vector of OLS residuals.

(e) Now run this program with the command:

```
>> results = ols_ts(y0,X0,info);
```

(f) Notice first that the screen output shows regression results (betas, p-values, etc.) that are virtually identical with those obtained in JMP. Notice also that “**Temp**” in **info.names** above was used to assign a name to the dependent variable on the second line of the screen output, and the additional elements of **info.names** were used to assign names to the dependent variables in subsequent lines. [So the main advantages of doing regressions in JMP are its *graphical* outputs along with a host of additional diagnostics.]

(g) Next, you will store the desired residuals as a vector, **res_ols**, in the workspace by writing:

```
>> res_ols = results.resid;
```

Again, you should check to be sure that these are the same as those just constructed in MATLAB.

- (h) You are now ready to fit a spherical variogram to these residuals, as in past assignments. To do so, define the combined matrix, **M**, of measurement locations and OLS residuals by;

```
>> M = [L0,res_ols];
```

and then fit a spherical variogram with the command:

```
>> var_spher_plot(M);
```

where defaults are used for maxdist and binmax.

Notice first that this variogram is rather “shallow”. Comment on this in terms of the associated *relative nugget effect* (nugget/sill). How does this relate to the nearest-neighbor residual regression above?

- (i) Note finally that in spite of its shallow nature, this spherical variogram still provides enough additional *covariance information* to allow a more refined estimation of this temperature trend surface than is obtainable from the OLS regression above.
- (5) To construct these more refined estimates, you will now employ *geostatistical regression*, as operationalized in the MATLAB program, **geo_regr**. By opening this program in MATLAB, you will see that we have already constructed all of the necessary inputs.
- (a) All that is required is to construct the list of explanatory variable names, **vnames**, by removing “**Temp**” from **info.names** above. This can be accomplished with the command:

```
>> vnames = info.names(2:end,:)
```

As usual, by leaving off the semicolon, the resulting screen output should confirm that **vnames** has been defined successfully. [Of course you could define it directly with a new “**strvcat**” command.]

- (b) Now run the geostatistical regression with the command:

```
>> geo_regr(y0,X0,L0,vnames);
```

- (c) Given the screen output of this regression, notice first that the parameters for the variogram estimated are slightly different, and in particular that there now appears to be slightly *more spatial dependency* in the new regression residuals than was detected in step 4(i) above. Discuss this difference in terms of the new *relative nugget effect*, and comment on why one might expect such a difference. [HINT: Recall the *iterative* nature of the estimation procedure in **geo_regr**.]
- (d) Next consider the beta values in this screen output. By comparing them with the **OLS** regression screen output in step 4(e) above, you should see that there is actually very little difference in the beta estimates. Comment on why this might be so. [HINT: Recall the conditions under which OLS beta estimates are unbiased.]
- (e) Given these beta estimates, comment on the overall spatial pattern of 1981 temperatures in England, and in particular on the nonlinear effect, \mathbf{x}^2 .
- (f) Note finally that in spite of the similarities between their beta estimates, there is noticeable *systematic difference* in the significance levels between **OLS** and **geo_regr**. Discuss the reasons why one might expect this, given the variogram results in step 4. [HINT: Consider what the lower relative nugget effect indicates.]
- (6) Given these preliminary analyses, you are now ready to perform *geostatistical kriging* of this temperature data in MATLAB using the program **geo_krige.m**. To do so, open this program in MATLAB and observe that the first three inputs (**y0,X0,L0**), based on the observed data at measurement points, have already been constructed above. What remains to be constructed is a set of locations, **L1**, where predictions are to be made, along with comparable values of each explanatory variable, **X1**, at each location. (The bandwidth, **h**, to be used for the kriging predictions will be considered below.) But rather than considering only a single representative point (as in past assignments), you will now construct a *full grid* of points to be used for interpolating a full prediction surface (in a manner similar to universal kriging in Geostatistical Analyst).
- (7) To construct such a grid, it is useful to start by examining the extent of the measurement data in ARCMAP. If you examine the coordinates on the England map you will see that the boundary is contained in a box with base interval [160,580] and height interval [120,580]. (Recall that we wish to *avoid extrapolation* whenever possible.) To construct the desired grid, you will use the MATLAB program, **grid_form.m**. As usual, begin by opening this program open MATLAB.
- (a) The **box limits** and **cell size** are required as inputs to this program. To define the box, set **Xmin = 160, Xmax = 580, Ymin = 120, Ymax = 580**.
- (b) Also observe from the map that a reasonable **cell size** here is 20x20 km. So in the program you will set **Xcell = 20, Ycell = 20**. To construct the grid, write
- » **G = grid_form(Xmin,Xmax,Xcell,Ymin,Ymax,Ycell);**

You should see the upper left corner of the grid displayed, just as a visual consistency check. [Be sure that the figure window is *closed* before executing this command, so that the grid is not confounded with the previous variogram figure.] The matrix **G** contains the grid point coordinates at which the kriging is to be made.

(c) To construct the appropriate inputs **X1** and **L1** for **geo_krige** write

```
» x = G(:,1);
» y = G(:,2);
» X1 = [x,y,x.^2];
» L1 = [x,y];
```

(Note that **x** and **y** will automatically be redefined by this operation, so there is no need to clear the old values of **x** and **y** above.)

(8) You are now ready to do the kriging. If the number of explanatory variables is denoted by **k** (= 3), then the usual rule of thumb here is that the kriging for each point should involve at least **k+1** (= 4) data points. Observe from the map displayed in ARCMAP that a bandwidth of **h = 50 km** is big enough to ensure that neighborhoods of this size will contain at least 4 data points for most grid points. [The program automatically chooses the **k+1** closest data points when this condition is not met.] So to run the desired kriging, write:

```
» DAT = geo_krige(y0,X0,L0,X1,L1,50);
```

(a) When completed, you can view (and save) the vectors of GLS-regression coefficients and corresponding P-Values as a matrix, **M**, by writing (be sure to set **Numerical Format** = "short g" using **File → Preferences**):

```
» M = DAT{1}
```

Also, you can view (and save) the vector, **p**, of estimated variogram parameters by writing:

```
» p = DAT{2}
```

Compare these results with the screen output for **geo_regr** above, and comment on why this relationship is to be expected.

(b) Recall that the **sill value**, *s*, should be an estimate of the true variance of the residuals in this regression. Hence it is of interest to compare this estimate with the standard (**OLS**) estimate of variance, namely *Mean Square Error (MSE)*:

$$MSE = \frac{1}{n-(k+1)} \sum_{i=1}^n \hat{\epsilon}_i^2$$

where $(\hat{\epsilon}_i : i = 1, \dots, n)$ are the **OLS** regression residuals. To do so, you can construct this quantity in MATLAB as follows:

1. First to construct the appropriate data matrix, now add the *unit vector*, **u**, to the data matrix, **X0**, by writing:

```
» u = ones(48,1);
» XX0 = [u,X0];
```

2. In terms of this augmented data, recall that the **OLS** estimates of the betas are given by

```
» b0 = inv(XX0'*XX0)*XX0'*y0;
```

3. Hence the **OLS** residuals above can be calculated as

```
» res = y0 - XX0*b0;
```

[so that **res** = $(\hat{\epsilon}_i : i = 1, \dots, n)'$].

4. One can compute the **MSE** for this regression (with $k = 3$ explanatory variables) by writing:

```
» MSE = (1/(48 - 4))*sum(res.^2);
```

5. As one final comparison, consider the simple *sample variance* estimate

$$VAR_n = \frac{1}{n-1} \sum_{i=1}^n (\hat{\epsilon}_i - \bar{\epsilon})^2, \text{ where } \bar{\epsilon} = \frac{1}{n} \sum_{i=1}^n \hat{\epsilon}_i$$

6. This can be computed in MATLAB by the single command:

```
» VAR = var(res);
```

7. How do these **MSE** and **VAR** estimates compare with the sill value? How does this relate to the discussion of variance estimation in Section 4.10 of Part II (Continuous Data Analysis) in the NOTEBOOK? [Recall that variance is obtained from the covariogram, $C(h)$, at $h = 0$.]

8. What additional information about this relation is provided by your discussion of *spatial dependence* in part 5(c) above?

- (9) The next objective is to import the Kriging and Std Error results (**DAT{3}** and **DAT{4}**) along with the 528 grid points to ARCMAP. To do so, write:

- » **K1 = DAT{3};**
- » **S1 = DAT{4};**
- » **K(:,1:2) = L1;**
- » **K(:,3) = K1;**
- » **K(:,4) = S1;**

The (528x4) matrix **K** now contains all the desired data (**L1** contains the grid points). To save this as a text file, **krg_data.txt**, in your home directory, write

- » **save S:\home\krg_data.txt K -ascii**

- (10) To be sure that the data is in proper format, it is best to import it to EXCEL.
- (a) (i) In EXCEL [version 2003] you can change the data format to a more visually appropriate form by first selecting all four columns, clicking **Format** → **Cells** → **Number**, and then setting **Decimal places** = 3.
 - (ii) In EXCEL [version 2007] you can change the data format to a more visually appropriate form by first selecting all four columns, clicking **Home** and moving to the “Cells” box. Here click **Format** → **Format Cells** (at bottom of list). Then in the “Format Cells” window click **Number** → **Decimal places** = 3.
 - (b) Now select the top row and insert a new row by clicking **Insert** → **Rows**.
 - (c) Add column labels, **X**, **Y**, **Krige**, **Std_Err**, in this new (top) row.
 - (d) To save this data in appropriate form, click **Save as** → **Text(tab delimited)**.
 - (e) Finally, change the extension of this file to **.tab** (which ARCMAP recognizes as a text file in tab format). You are now ready import to import **krg_data.tab** to ARCMAP.
- (11) In ARCMAP click on the **Add Data** button and add **krg_data.tab** from your home directory. It will appear as a new layer in the Table of Contents.
- (a) Right click on the layer “krg_data.tab” and click **Display XY Data**.
 - (b) The **X Field** and **Y Field** should already be set to “X” and “Y”.
 - (c) Click **OK** and the grid of points should now appear on the map and as a new layer “krg_data.tab Events”.

- (d) Save as a shape file, **krg_data.shp** using **Data** → **Export Data**, and add it to the data frame. (You can now remove “krg_data.tab” and krg_data.tab Events”.)
 - (e) If you drag this layer to a position just below the “mask” layer then only the points inside the England boundary should be visible.
 - (f) Before proceeding further, it is best to turn off the display of these points.
- (12) Next, you will interpolate these kriged values using a **raster spline**. Here you will use the same procedure as in section (d) of Study 2 in Assignment 3. Hence the following instructions focus mainly on the new features of this spline interpolation.
- (a) In the **Spline** window be sure that **Input points** = “krg_data”, and set **value field** = “Krige”. Also set **Number of points** = 4, and leave other values at defaults. [The small number of points together with the small value of **Weight** (default = 0.1) will yield an interpolation that closely fits the kriged data points.]
 - (b) At the bottom of the window click **Environments** → **Processing Extent**. Then set **Extent** = “Same as layer eng_temp_bnd”. This will cover all of the map.
 - (c) To make this into a permanent file click **Data** → **Export Data**, In the “Export Raster Data” window set **Name** = “krige_spline” and **Format** = “GRID”. Click **Save**, and export to the map as a layer
 1. Click **OK**, and a new layer “krige_spline” will appear in the Table of Contents. To view this result properly, you should drag the layer “krige_spline” to a position between “mask” and “eng_temp_bnd”.
 2. If the image shows colors, proceed to the next step. If the image appears in black-and-white and looks “smooth”, you can alter this by right clicking on “krige_spline”, and then clicking **Properties** → **Symbology** and changing the setting of **Show** to “Classified”.
 3. You can now choose the interval size (try **Classes** = 20) and color ramp to be whatever you like. (To reverse the color order, right click on any symbol in the **Layer Properties** window, and select **Flip Colors**.)
- (13) Now, repeat step (12) for the standard error values, **Std_Err**, in layer “krg_data”.
- (a) Save the raster spline file as **stderr_spline** in your home directory.

(b) When you display these results using the default classification settings, you will see that the relevant standard deviation values lie in a *narrow range*, so the most of the map has the *same color*.

1. To improve this representation, on the right-hand side of the **Layer Properties** window, click **Classify...** and you will see that the default classification scheme (usually “Natural Breaks” or “Equal Intervals”) leaves most of the color histogram in the same interval.
2. Try the “Quantile” representation in this case. You will see that the representation is improved. You can then use “Manual” to make finer adjustments to color classes if you like.
3. Also try different settings of **Classes**.

(14) Given this construction, you are now ready to do some **ANALYSIS**.

- (a) First, zoom in on the area around the point (338,402), which is just about the center of Liverpool (You can also use the **Go to XY** tool on the **Tools** toolbar.)
 1. Using the Identify tool, with **krige_spline** displayed, determine the kriged estimate of (Celsius) temperature in Liverpool. (In the “Identify” window you may need to set **Identify from** = <all layers> in order to see the value for **krige_spline**.)
 2. Now repeat this procedure with **stderr_spline** displayed and determine the standard error of this kriged estimate.
 3. Using these values, determine a 95% prediction interval for the mean 1981 temperature in Liverpool.
 4. Given that Fahrenheit (F) is related to Celsius (C) by $F = (9/5) \cdot C + 32$, determine a 95% prediction interval for Liverpool temperature in Fahrenheit.
 5. The actual mean temperature for August in Liverpool is about $61^\circ F$.² How does this compare with your result above?
- (b) To compare this with an *exact* estimate, return to MATLAB and redo the krige for this single point.
 1. Here the only change required is the definition of **X1** and **L1** in (7).(c) above.

² This is taken from the *Washington Post Archives*. See the file **Liverpool_Historical_Data.pdf** in the **Eng_Temp** directory.

2. Let $x = 338$, $y = 402$, and again set $L1 = [x,y]$ and $X1 = [x,y,x^2]$ (no period required for the scalar case).
 3. Now repeat the command in (8) above and examine **DAT{3}** and **DAT{4}**.
 4. How might you account for difference between this **standard error** value and the value obtained by the spline interpolation? (HINT: Look at the raster spline interpolation near Liverpool with the “mask” layer turned off.)
- (c) Next, display only the layers “eng_temp_81”, “mask”, “krige_spline”, “eng_temp_bnd” and zoom in on the area around the **measurement sample point** (422,380). You will notice that this is point is exactly on the line between the kriged points, (420,380) and (440,380).
1. Now compare these two kriged values with the measured value at (422,380). Does anything seem strange to you?
 2. If so, how might you explain this apparent anomaly?