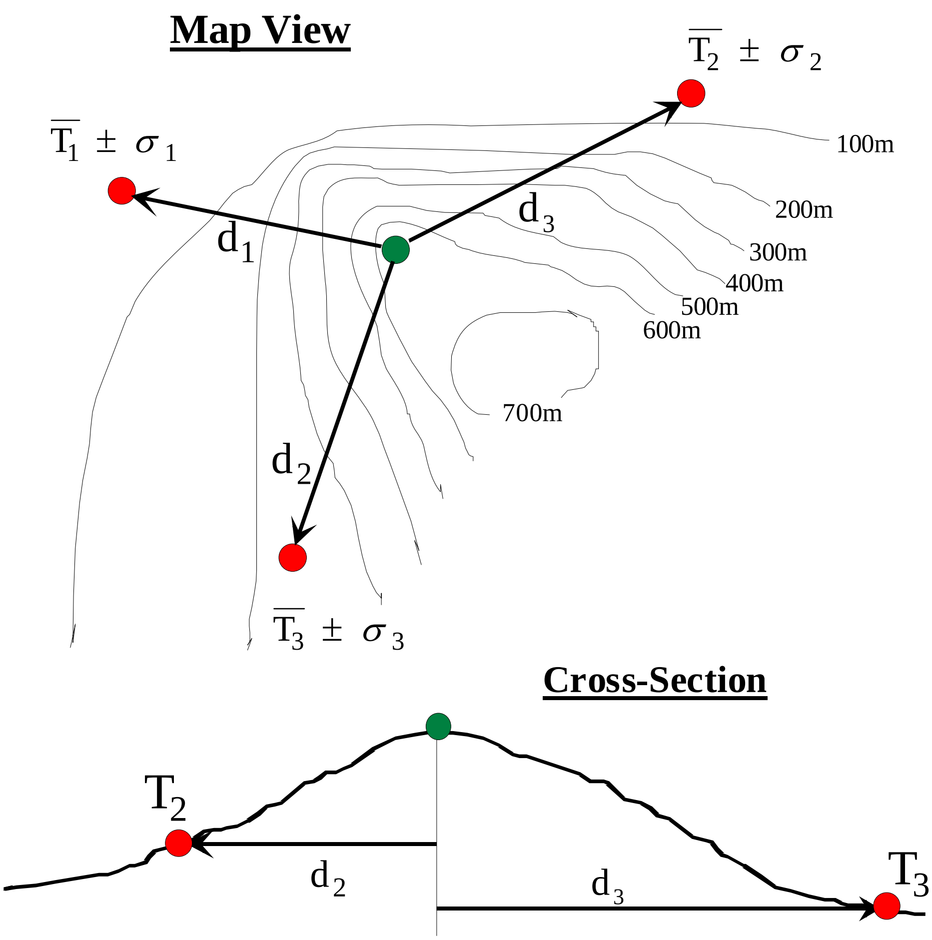
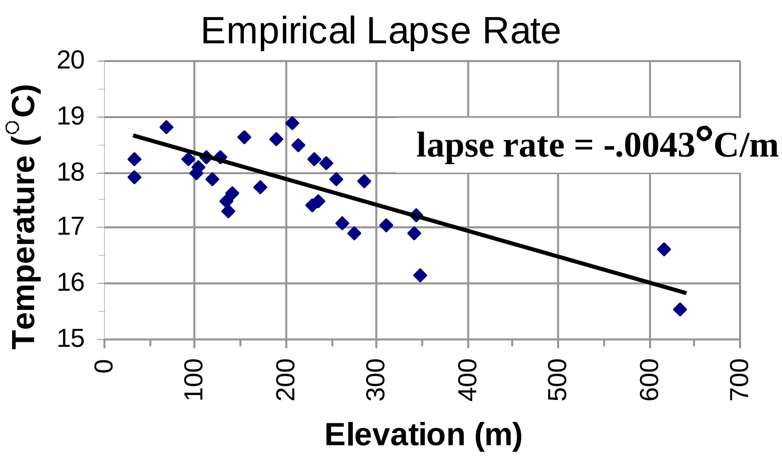
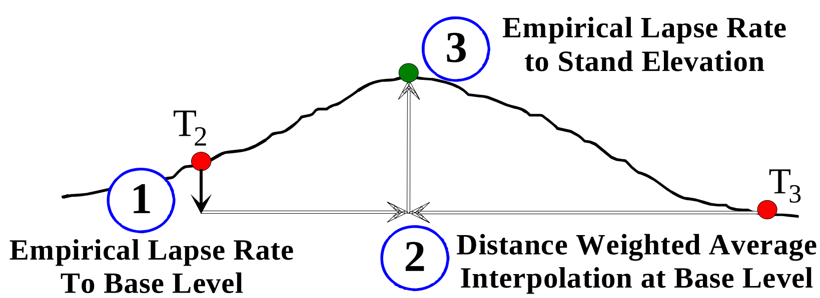
INTRODUCTION

In Earth science research, climatic conditions can have significant effects on dynamic processes. For example, the growth of forests is affected by temperature, rainfall and other climatic variables. However, these climate parameters are rarely measured at the forest stands whose growth is being investigated. The climate conditions are measured at nearby weather stations, though, and it is a common approach to use the weather station data to 'predict' the climate at the study site. This situation is illustrated in the sketches below. Three weather station locations are shown in red, each with a known mean annual temperature $\overline{T}$ and associated variance σ^2, as determined from measurements made over some historical period. The study site, shown in green, is where we would like to know the mean annual temperature, but no temperature measurements have ever been recorded there. The straightforward objective is to find the best approximation possible to the study site temperature given the weather station data.



It is evident from the cross-section that a simple, distance-weighted average of the weather station data is inadequate to describe the temperature at the study site, since temperatures depend on elevation and other topographical features. One approach to remedy this situation might utilize the following strategy. If the dependence of temperature on elevation were known, for instance from a regression of weather station temperatures on weather station elevations, the effect of elevation could be accounted for by performing the spatial interpolation at some base elevation. A three-step interpolation procedure could be implemented as shown in the sketches below. By modeling how temperature changes with elevation, the variability in temperature caused by elevation can be removed before the spatial interpolation is performed. In the graph shown below, a simple linear regression is used to determine the empirical lapse rate. This lapse rate is used to translate the station temperatures to some 'base level' elevation, where spatial interpolation is performed. The interpolated temperature at the base level is then re-translated to the study site elevation using the same empirical lapse rate.





With this procedure, the spatial interpolation is performed while taking elevation into account. But while this is certainly an improvement over a simple distance-weighted average of station data, it does not account for other independent factors which also affect temperature. Further, a natural question to ask would be whether or not distance weights are the optimal interpolation weights to use. These concerns provide motivation to explore the generalization of the above three-step interpolation process.

In conceptual terms, the above interpolation process seeks to account for temperature variability which is explainable by linear regression on elevation, and then spatially interpolate the *residuals* by a distance-weighted average. When viewed in this light, two improvements to the process are immediately apparent:

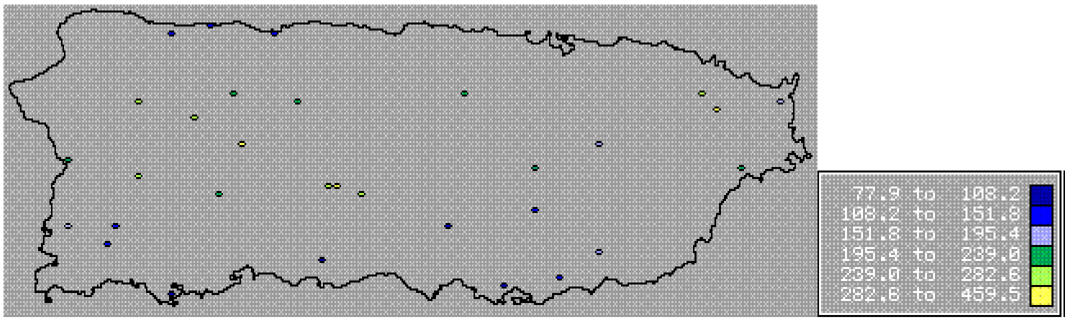
(1) Use a more general regression procedure, and try to explain as much of the variability in temperature as possible using combinations of several independent factors which affect temperature.

(2) Use *optimal* weights for the spatial interpolation of residuals.

This generalization is the method of Kriging. The objective here is to use Kriging to produce a 'continuous' map of rainfall over the land surface area of Puerto Rico. The available dataset is given in the table below.

Station Label	Latitude (°N)	Longitude (°W)	Elevation (m)	Aspect (degrees True)	Center Direction (degrees True)	Center Distance (km)	Sea Distance (km)	Average Annual Rainfall (mm)
s1	17.967	66.233	3.0	154.9	141.3	36.3	1.82	108.2
s2	18.467	66.717	15.2	62.4	311.4	43.3	0.46	148
s3	18.35	66.800	320.0	288.4	290	45.4	14.81	216.1
s4	18.083	67.150	76.2	295	259.5	85.1	4.07	167.2
s5	18.250	66.033	76.2	90	85.2	46.9	21.12	158
s6	18.483	66.850	21.3	294.9	302.6	57.1	0.46	124.9
s7	18.117	66.167	426.7	0	110.6	33.1	18.15	149.2
s8	18.200	66.167	426.7	180	93.6	31.2	25.92	198.4
s9	18.083	66.350	61.0	274.3	148	17.7	12.26	108
s10	18.350	66.317	121.9	180	41.2	20.1	13.58	199.4
s11	17.950	66.933	3.0	209	241.9	65.5	21.46	77.9
s12	18.333	65.650	12.2	303.7	81.5	92.6	2.05	165.9
s13	17.983	66.117	61.0	180	126.4	45.2	3.37	138.3
s14	18.150	66.833	27.4	256.5	260.5	47.4	17.53	216.8
s15	18.167	66.583	609.6	238.3	251.6	18.4	21.6	303.8
s16	18.167	66.600	762.0	229.6	253.3	20.3	21.22	282.6
s17	18.317	65.783	701.0	38.6	81.3	76.9	8.13	459.5
s18	18.050	67.067	30.5	132.7	255.3	76.3	8.35	120.9
s19	18.300	66.883	365.8	33.7	280.5	53.5	20.38	248.6
s20	18.183	67.000	457.2	346	266.6	66.4	17.17	282
s21	18.217	67.150	24.4	180	270	83.8	1.3	204.2
s22	18.200	65.733	39.6	180	91.4	81.9	2.01	214.7
s23	18.033	66.033	61.0	88.7	114.6	51	6.28	168.1
s24	18.017	66.617	12.2	202.8	222.5	30.9	4.38	91.4
s25	18.467	66.933	15.2	123.7	296.6	64.8	2.05	138.3
s26	18.333	66.667	60.4	110.5	296.6	30.2	15.83	204.1
s27	18.150	66.533	685.8	117.9	236.3	13.8	18.67	274.7
s28	18.250	66.783	152.4	305	275.4	41.1	25.18	339.5
s29	18.350	65.817	106.7	270	77.8	73.7	6.42	240.8
s30	18.083	67.050	106.7	275.3	257.8	73.6	12.17	143.5
s31	18.333	67.000	106.7	258.2	281.6	67.6	17.5	246.6

The data in the table were collected from various sources, and values in some of the columns were calculated in Idrisi. More data is on the way, but my contacts in Puerto Rico have been slow in getting it to me. The map below (displayed in InfoMap) shows average annual rainfall in millimeters at the weather station locations listed in the above table. The base file ‘prtoric2.bdy’ and associated data file ‘rainfall.dta’ used to create the image below are provided on the disk accompanying this project.



Kriging is performed in InfoMap, but the values at intermediate locations are only displayed on the screen and are not retrievable. The problem here might be one of location: InfoMap base files only store location information for the sites where raw data is available. At places where there are no data, InfoMap base files contain no location information. So, in order to save the Kriged data values within InfoMap, a location must be specified at each place where an interpolated value is desired. This *could* be done by modifying the base file, adding all the locations at intermediate points, but I will defer that to a later time, and work with images in Idrisi for now. The end results will be easily imported into InfoMap, as the data format is nearly identical. The InfoMap file will be required to obtain variogram model parameters for computation of the covariance matrix.

KRIGING

The overall objective is to estimate the unknown value of rainfall at intermediate locations based on empirical data at fixed locations. Using the notation from class, let X(*s*) represent the rainfall at any single location, *s*, within the study area. The raw data can then be written in the form of ordered pairs

$$\{(s_1, X_1),(s_2, X_2), \ldots ,(s_n, X_n)\}$$

where each s_i, i = 1 . . n, represents the location of the i^{th} weather station, and X_i, i = 1 . . n, is the average annual rainfall measured at that station. Then, the familiar model for rainfall at any weather station location is given by

$$X(s_i) = \mu(s_i) + U(s_i)$$

where $\mu(s_i)$ represents the best estimate of the rainfall found from regression on independent parameters at location s_i, and $U(s_i)$ is the regression residual at s_i. Vector notation for i = 1, . . ,n is written here as

$$\mathbf{X}(\boldsymbol{s}) = \boldsymbol{\mu}(\boldsymbol{s}) + \mathbf{U}(\boldsymbol{s})$$

where the bold-type $\mathbf{X}(\boldsymbol{s}) = \{X(s_1), X(s_2), \ldots , X(s_n)\}^T$ indicates a column vector of dimension n.

REGRESSION ON INDEPENDENT PARAMETERS

Now, call the independent parameters, such as elevation, aspect, etc., $\mathbf{Y}_1(\boldsymbol{s})$, $\mathbf{Y}_2(\boldsymbol{s})$, . . . , $\mathbf{Y}_k(\boldsymbol{s})$. Note that these are also column vectors. Then, the regression component $\mu(\boldsymbol{s})$ is found by determining the (least-squares) regression coefficients β in

$$\mathbf{X}(\boldsymbol{s}) = \mu(\boldsymbol{s}) + \mathbf{U}(\boldsymbol{s})$$
$$= \mathbf{Y}(s)\beta + \mathbf{U}(\boldsymbol{s})$$

where $\mathbf{Y}(s)$ is the n by k+1 matrix $[\mathbf{1}, \mathbf{Y}_1(\boldsymbol{s}), \mathbf{Y}_2(\boldsymbol{s}), \ldots, \mathbf{Y}_k(\boldsymbol{s})]$, and β is a k+1 by 1 column vector of regression coefficients. This is a general linear model, and the aim is that the regression explains all of the *regular* variability in the data $\mathbf{X}(\boldsymbol{s})$. If this is attained, then the residuals $\mathbf{U}(\boldsymbol{s})$ will display no pattern, i.e., $\mathbf{U}(\boldsymbol{s})$ will be covariant stationary, with covariance structure $\text{cov}[\mathbf{U}(\boldsymbol{s})] = \sigma^2\mathbf{I}$. This property of residuals is required for Kriging. However, in the general case, there is no guarantee that the residuals will be covariant stationary. So, the general linear model $\mathbf{X}(s) = \mathbf{Y}(s)\beta + \mathbf{U}(s)$ must first be transformed into a classical model $\mathbf{X}^*(s) = \mathbf{Y}^*(s)\hat{\beta} + \mathbf{U}^*(\boldsymbol{s})$ through the transformation

$$(\mathbf{Y},\mathbf{X}) \rightarrow L^{-1}(\mathbf{Y},\mathbf{X})$$

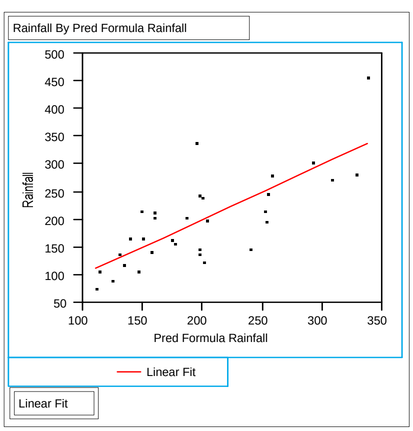
where the matrix L^{-1} is the inverse of the lower triangular Cholesky factor of the residuals covariance matrix:

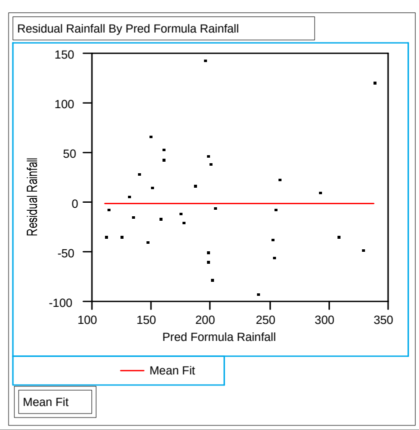
$$L\, L^T = \text{cov}(\mathbf{U})$$

Then, the subsequent regression on the transformed variables $(\mathbf{Y}^*,\mathbf{X}^*)$ yields $\text{cov}[\mathbf{U}^*(\boldsymbol{s})] = \sigma^2\mathbf{I}$. The coefficients $\hat{\beta}$ are then used as estimators of the classical linear regression coefficients in

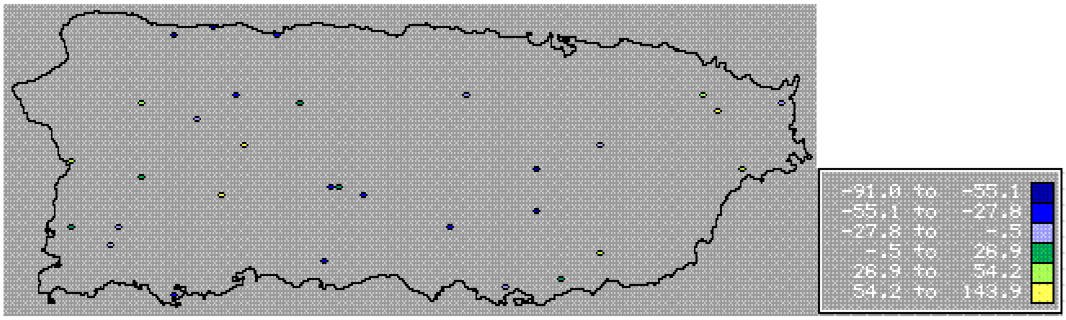
$$\mathbf{X}(\boldsymbol{s}) = \mathbf{Y}(s)\hat{\beta} + \mathbf{U}_1(\boldsymbol{s})$$

where the subscript on $\mathbf{U}_1(\boldsymbol{s})$ denotes a first iteration. Since $\mathbf{U}_1(\boldsymbol{s})$ may still not be covariant stationary, further iterations of this process can be carried out, until the covariance structure of $\mathbf{U}_j(\boldsymbol{s})$ is sufficiently close to $\sigma^2\mathbf{I}$ form. The 'Universal Kriging Transform Program' listed at the end of this paper was written to carry out this transformation. For this project, the regression was performed in JmpIn, and the results of the general linear model are shown below:

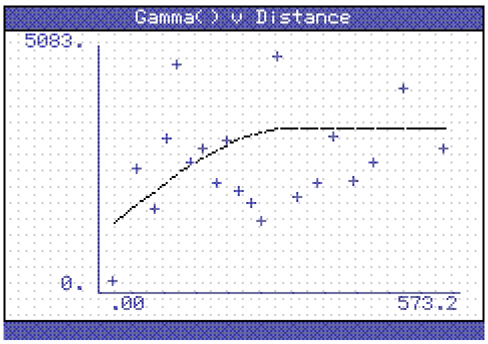




Response:	Rainfall	Parameter Estimates				
RSquare	0.583508	Term	Estimate	Std Error	t Ratio	Prob>|t|
RSquare Adj	0.553758	Elevation	0.2401363	0.042049	5.71	<.0001
Root Mean Square Error	54.56738	Latitude	159.76765	66.47986	2.40	0.0231
Mean of Response	198.0516	Intercept	-2756.746	1210.156	-2.28	0.0306



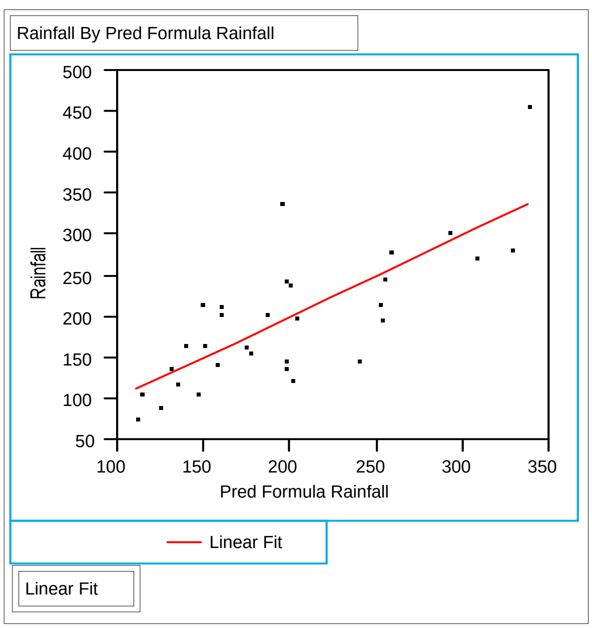
From this model, the residuals at each location were re-calculated in InfoMap and displayed above. The following variogram was then calculated from the residuals:

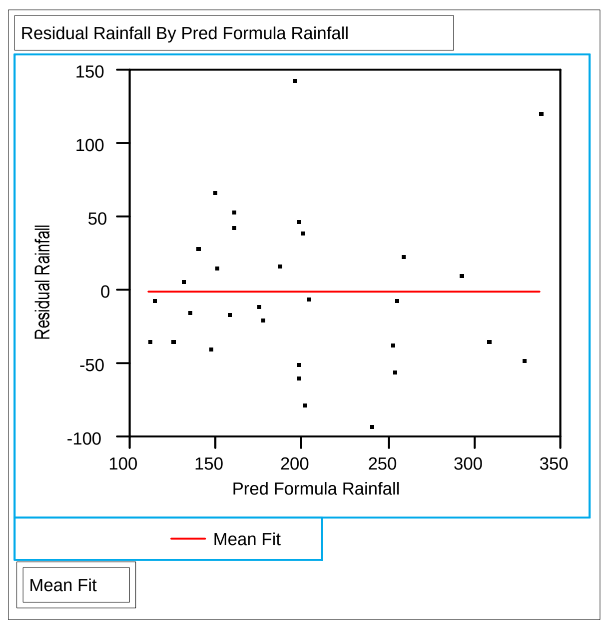


For this Spherical variogram model calculated in InfoMap, the following parameters were obtained:

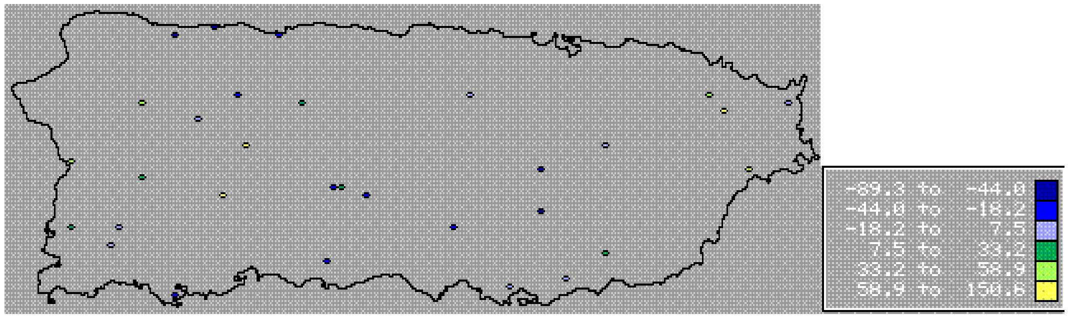
Nugget:	1322
Range:	309.5
Sill:	3456

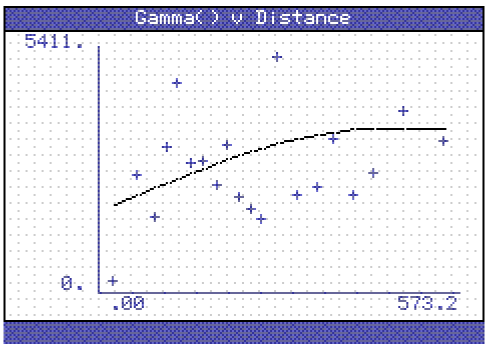
With the above variogram parameters, the first iteration residuals $\mathbf{U_1}(\boldsymbol{s})$ were calculated using the Universal Kriging Transform Program, and the results are plotted below:





The spatial distribution of the first iteration residuals is shown below, along with the corresponding variogram, calculated in InfoMap.





For the Spherical variogram model of the first iteration residuals, the following parameters were obtained:

Nugget: 1322
Range: 309.5
Sill: 3456

Examination of the residuals shows that the first iteration produced changes the residual values at all station locations, as shown in the following table:

Station Label	First Residual	Second Residual		Station Label	First Residual	Second Residual
s1	-6.32009	-10.597		s17	121.4461	131.1056
s2	-49.3336	-34.2563		s18	-13.4846	-14.5344
s3	-35.7343	-25.0126		s19	-6.24415	2.568651
s4	16.56889	16.82176		s20	23.90021	28.23723
s5	-19.3123	-12.5974		s21	44.59908	50.00787
s6	-76.4547	-60.7549		s22	54.16506	58.9246
s7	-91.031	-89.265		s23	29.10734	27.41692
s8	-55.0917	-50.114		s24	-33.3177	-35.6548
s9	-38.981	-38.7367		s25	-59.0336	-43.9563
s10	-4.8633	5.746818		s26	17.32113	27.23879
s11	-33.904	-38.8388		s27	-33.0226	-29.8337
s12	-9.3043	0.586197		s28	143.8893	150.6471
s13	7.295722	3.670563		s29	40.18677	50.78833
s14	67.1831	70.00103		s30	-14.4553	-14.1852
s15	11.65972	15.46349		s31	48.70282	58.64657
s16	-46.137	-42.2474				

The iterations on the residuals should be repeated until the variogram displays covariant stationary properties, i.e., until the variogram flattens out. For the purposes of this project, I will stop at one iteration, and use the variogram parameters listed above to compute the covariance matrix used in Kriging below.

SPATIAL INTERPOLATION OF RESIDUALS BY KRIGING

The Kriging estimate for unknown rainfall at any location $s \notin \{s_1, \ldots, s_n\}$ is found by first applying the regression model, and then spatially interpolating the residuals:

$$\begin{aligned} X(s) &= \mu(s) + U(s) \\ &= Y(s)\hat{\beta} + \hat{\mathbf{U}}(s) \end{aligned}$$

where the $\hat{\beta}$ regression coefficients are determined from the final iteration of the regression model (which also yields $\mathbf{U_j(s)}$), and where $\hat{\mathbf{U}}(s)$ is estimated by the following expression:

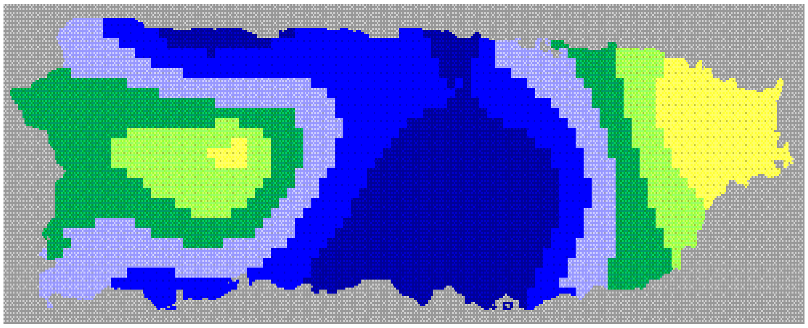
$$\hat{\mathbf{U}}(\mathbf{s}) = \sum_{\mathbf{i}=1}^{\mathbf{n}} \lambda_{\mathbf{i}}(\mathbf{s}) \cdot \left(\mathbf{U}_{\mathbf{j}}(\mathbf{s}_{\mathbf{i}})\right)$$

That is, the estimated residual at the unknown location s is a weighted average of the residuals at the known locations. The optimal weights $\boldsymbol{\lambda}^*(s)$ are determined from the covariance matrix of residuals, by

$$\boldsymbol{\lambda}^*(s) = \mathbf{C}^{-1}\mathbf{c}(s)$$

where $\mathbf{C} = \text{cov}[\mathbf{U_j}(\boldsymbol{s})]$, and $\mathbf{c}(s) = (\sigma_{1s}, \ldots, \sigma_{ns})$, the covariance between the unknown location and the station locations, which are set to zero if the station location is outside the Kriging bandwidth.

The Kriged residual map $\hat{\mathbf{U}}(s)$ is shown on the following InfoMap output.



Obviously, there is still some structure to these residuals, and a few more iterations in the regression model are warranted. Unfortunately, the program which accomplishes the spatial interpolation is not finished yet, so the term project ends with this map.

SPATIAL INTERPOLATION OF RESIDUALS BY KRIGING

In the first part of this paper, a regression model was formulated which, through several iterations produced residuals which are covariant stationary. The remaining task is to interpolate the residuals to form a continuous map of temperature values. The following passage reviews the general procedure. The Kriging estimate for unknown rainfall at any location $s \notin \{s_1, \ldots, s_n\}$ is found by first applying the regression model, and then spatially interpolating the residuals:

$$
\begin{aligned}
X(s) &= \mu(s) + U(s) \\
&= Y(s)\hat{\boldsymbol{\beta}} + \hat{\mathbf{U}}(s)
\end{aligned}
$$

where the $\hat{\boldsymbol{\beta}}$ regression coefficients are determined from the final (j^{th}) iteration of the regression model (which also yields $\mathbf{U_j}(\boldsymbol{s})$), and where $\hat{\mathbf{U}}(s)$ is estimated by the following expression:

$$\hat{\mathbf{U}}(\mathbf{s}) = \sum_{\mathbf{i}=1}^{\mathbf{n}} \boldsymbol{\lambda}_{\mathbf{i}}(\mathbf{s}) \cdot \left(\mathbf{U}_{\mathbf{j}}(\mathbf{s}_{\mathbf{i}})\right)$$

That is, the estimated residual at the unknown location s is a weighted average of the residuals at the known locations. The optimal weights $\boldsymbol{\lambda}^*(s)$ are determined from the covariance matrix of residuals, by

$$\boldsymbol{\lambda}^*(s) = \mathbf{C}^{-1}\mathbf{c}(s)$$

where $\mathbf{C} = \mathrm{cov}[\mathbf{U_j}(\boldsymbol{s})]$, and $\mathbf{c}(s) = (\sigma_{1s}, \ldots, \sigma_{ns})$, the covariance between the unknown location and the station locations. Since the covariance matrix is symmetric and positive definite, a Cholesky factorization is possible which allows the following simplification:

$$
\begin{aligned}
\boldsymbol{\lambda}^*(s) &= \mathbf{C}^{-1}\mathbf{c}(s) \\
&= (\mathbf{L}\mathbf{L}^T)^{-1}\, \mathbf{c}(s) \\
&= [(\mathbf{L}^T)^{-1}\, \mathbf{L}^{-1}]\, \mathbf{c}(s) \\
&= [(\mathbf{L}^{-1})^T\, \mathbf{L}^{-1}]\, \mathbf{c}(s) \\
&= (\mathbf{L}^{-1})^T\, [\mathbf{L}^{-1}\, \mathbf{c}(s)]
\end{aligned}
$$

In executing the method, the value of σ_{is} is set to zero if the i^{th} station location lies at a distance which is greater than the Kriging bandwidth. The result is that the i^{th} station then has no bearing on the interpolated value. An implementation of this method is given in the program code listed at the end of this paper, the Universal Kriging Transform Program. This program uses the method described above to interpolate from point data to every pixel in an image. A test data set is provided below to illustrate the functioning of the program.

To illustrate the method with a simple dataset, consider the following 10 by 10 matrix of values which represent the elevation in meters of some 10 km by 10 km patch of land:

0	0	8	20	30	60	50	45	30	20
0	2	16	30	60	70	60	50	40	30
1	18	28	50	60	75	70	60	50	40
3	25	40	60	80	85	80	70	60	50
5	30	55	70	93	100	90	80	70	60
6	25	60	75	91	90	80	70	60	50
7	24	60	75	85	80	70	60	50	40
8	23	50	60	80	70	60	50	40	30
9	22	40	50	60	60	50	40	30	20
10	20	30	40	45	50	40	30	20	10

Suppose that the average annual temperature, in degrees centigrade, is known at some of these locations, as shown in the following ‘matrix map’:

32							19		29
			15			13			
	21								
					10				
		14							
	26			11					16
						15			

The resultant data set is shown below, along with the following second order least-squares regression of temperature on elevation:

$$\text{Temp} = 32.91 - 0.474(\text{Elev}) + .00249(\text{Elev}^2)$$

Elev	Temp	X	Y	Residual
0	32	0.5	9.5	3.514373
20	29	9.5	9.5	4.964742
23	26	1.5	2.5	2.632298
25	21	1.5	6.5	-1.922670
30	16	9.5	2.5	-5.810070
40	15	40	15	-4.584890
45	19	45	19	0.527704
50	15	50	15	-2.359700
60	14	60	14	-1.134520
70	13	70	13	0.090665
80	11	80	11	0.315850
100	10	100	10	3.766219

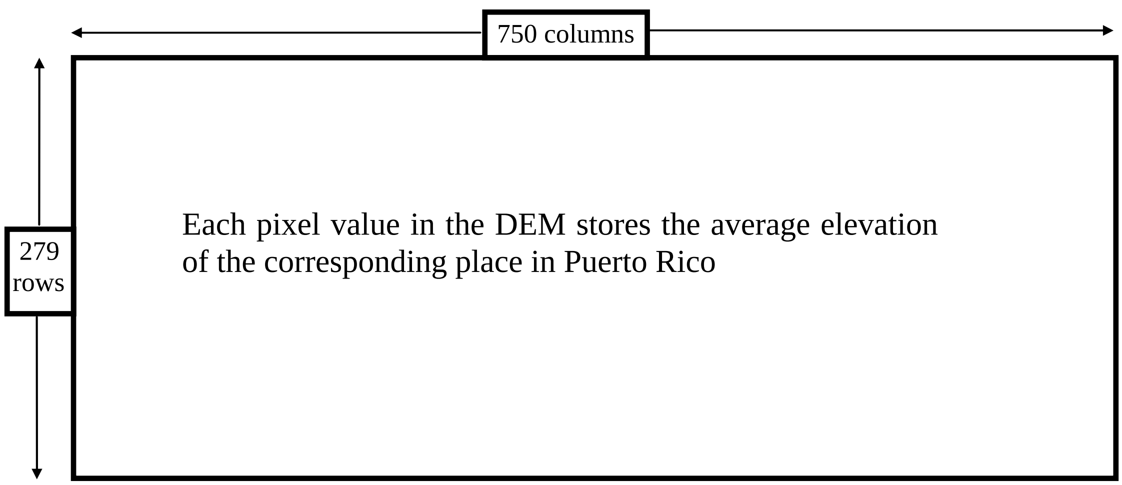
Modeling of these residuals in InfoMap gives the following spherical variogram model parameters:

nugget: 5.1
range: 7.37
sill: 11.5

Proceeding as if the regression residuals were covariant stationary, the Universal Kriging Interpolation Program, listed at the end of this paper, gives the following interpolated values, when a Kriging bandwidth of 5 is chosen:

30.8	30.4	24.3	22.1	19.6	13.1	16.0	18.0	23.1	29.0
30.8	29.5	23.4	18.7	11.8	11.7	14.0	17.1	20.1	23.3
30.3	22.9	19.4	13.6	12.1	11.2	12.5	14.9	17.2	20.3
28.4	20.7	16.4	12.0	9.8	10.4	11.2	13.3	14.8	16.9
29.5	19.9	13.6	11.2	9.7	9.8	10.7	11.4	12.6	14.2
30.1	22.7	13.5	11.5	10.4	10.5	10.8	11.6	13.2	15.6
30.1	23.6	14.0	11.8	10.8	10.9	11.7	12.7	14.9	17.6
30.0	24.4	16.4	14.2	11.6	11.9	12.9	14.5	17.1	20.4
29.8	25.0	20.7	16.4	13.9	13.2	13.8	17.1	20.0	23.7
29.4	25.7	22.1	18.8	17.0	15.5	17.4	20.0	23.6	27.5

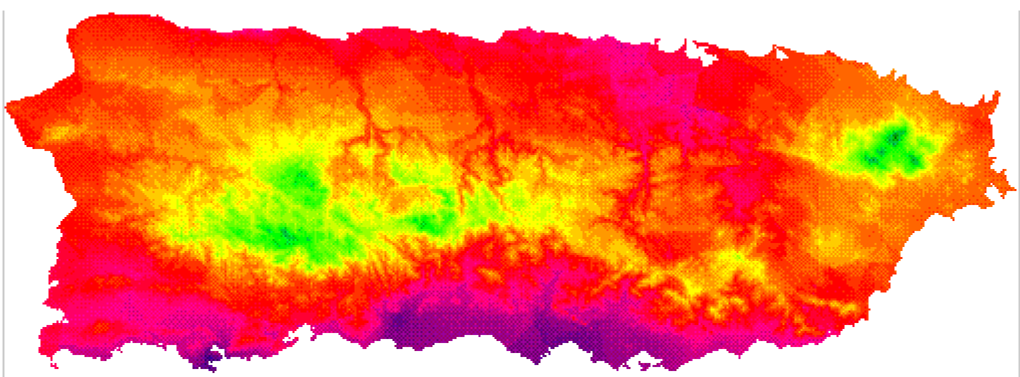
Applying this method to the Puerto Rico map, the US Geological Survey 1° Digital Elevation Model (DEM) is used as a basemap. This model consists of a digital image where the numerical value of each pixel represents the elevation at the corresponding location on the surface. The 1° DEM yields a resolution of about 30 meters, i.e., the size of each pixel in the image is about 30 meters on a side. In order to convert the DEM to fit into InfoMap, the pixel resolution was degraded to about 200 meters on a side, and the resulting image has the following characteristics:



So, generating a Kriged map of Puerto Rico Rainfall is no different than the sample dataset shown above, only the Puerto Rico image is somewhat bigger. The regression for Puerto Rico Rainfall was based on both elevation and latitude, as previously given. The (x,y) locations for the pixels are given by their latitude and longitude, and converted to the corresponding column and row numbers in the image, adjusted for location within the pixel to the nearest tenth pixel.

X	Y	Residual		X	Y	Residual
21.8	461.4	-10.597		185.3	661.0	131.1056
255.4	247.0	-34.2563		60.7	91.7	-14.5344
200.9	210.0	-25.0126		177.5	173.0	2.568651
76.3	54.7	16.82176		123.0	121.3	28.23723
154.2	550.1	-12.5974		138.6	54.7	50.00787
263.2	187.8	-60.7549		130.8	683.2	58.9246
91.9	491.0	-89.265		53.0	550.1	27.41692
130.8	491.0	-50.114		45.2	291.3	-35.6548
76.3	409.6	-38.7367		255.4	150.8	-43.9563
200.9	424.4	5.746818		193.1	269.1	27.23879
14.0	150.8	-38.8388		107.5	328.3	-29.8337
193.1	720.2	0.586197		154.2	217.4	150.6471
29.6	513.1	3.670563		200.9	646.2	50.78833
107.5	195.2	70.00103		76.3	99.1	-14.1852
115.2	306.1	15.46349		193.1	121.3	58.64657
115.2	298.7	-42.2474				

Using the regression on elevation and latitude, and the residuals shown in the table above, the Universal Kriging Interpolation Program was used to produce the Kriged Map of Puerto Rico shown below. The output is shown as an Idrisi image. In the image, green values are highest rainfall, grading through red to the lowest values of dark blue. Not that the interpolation accurately captures the rain forest areas of the island, and correctly displays very dry areas along the south coast.



```
      program krige
c*********************************************************************
c*            Universal Kriging Interpolation Program                *
c*********************************************************************
c  This program is designed to create an image file which displays    c
c  a continuous map of interpolated values calculated from point data c
c  by the method of Kriging.

      implicit none
      double precision C(50,50),sigma(50),u(50,3),sill,nugget,range
      double precision euclid,x1,x2,y1,y2,sphere,store(50,50),bndwdt
      double precision sum,dist,weight(50),regres,elev,y,w1(50)
      integer i,j,k,n,p,nrows,ncols,flag

c  Function Statements                                                c
      euclid(x1,y1,x2,y2)=sqrt((x1-x2)**2+(y1-y2)**2)
      sphere(x1,nugget,sill,range)=
     *              nugget+(sill-nugget)*
     *     (((3.0d0*x1)/(2.0d0*range))-((x1**3)/(2.0d0*(range**3))))
      regres(x1)=32.9142d0-.47408d0*x1+.00249d0*(x1**2)


c  Open Data Files                                                    c
      open(10,file='u.dat')
      open(11,file='basemap.img')
      open(12,file='krigemap.img')

c  Data Input Block                                                   c
      print*,'*****************************************************'
      print*,'*     Universal Kriging Interpolation Program         *'
      print*,'*****************************************************'
      print*

c  User Input: Image Size                                             c
      print*,'How many rows of pixels are there in the final image?   '
       read*,nrows
      print*,'How many columns of pixels are there in the final image?'
       read*,ncols
      print*

c  User Input: Variogram model parameters                             c
      print*,'This program will calculate the covariance matrix from  '
      print*,'spherical variogram model parameters.                   '
      print*,'Input value of the sill:                                '
       read*,sill
      print*,'Input value of the nugget:                              '
       read*,nugget
      print*,'Input value of the range:                               '
       read*,range
      print*

c  User Input:  Kriging Bandwidth
      print*,'When Kriging is performed, only point data within the   '
      print*,'Kriging bandwidth will be considered while interpolation'
      print*,'is performed.                                           '
      print*,'Input value of the Kriging bandwidth:                   '
       read*,bndwdt
      print*

c  File Input: Read in the point data in (x,y,u(x,y)) format          c
      print*,'The program reads the point data from a file            '
      print*,'called "u.dat", with data in the form (x,y,u(x,y))      '
      print*,'The program also reads the base map from a file         '
```

```
      print*,'called "basemap.img", in column format.                 '
      print*,'Are "u.dat" and "basemap.img" ready?                    '
      print*,'  1<enter>=continue   2<enter>=exit                     '
       read*,flag
      print*
      if(flag.ne.1)then
         print*,'Create data files and rerun.  Program Terminates.    '
         stop
      endif
      i=1
5     read(10,*,end=6,err=999) (u(i,j),j=1,3)
      i=i+1
      goto 5
6     n=i-1
      if(n.gt.50)then
         print*,'n= ',n,' > 50, too big for this program.'
         print*,'Program Terminates.'
         stop
      endif

c  Create Covariance Matrix from Variogram Model Parameters           c

c   Compute distance matrix                                           c
      do 15 i=2,n
         do 10 j=1,i-1
            store(i,j)=euclid(u(i,1),u(i,2),u(j,1),u(j,2))
            store(j,i)=store(i,j)
10       continue
15    continue
      do 20 i=1,n
         store(i,i)=0.0d0
20    continue

c   Compute Covariance Matrix                                         c
      do 35 i=1,n
         do 30 j=1,i
            if( (store(i,j).gt.0.0d0).and.(store(i,j).le.range) ) then
               C(i,j)=sill-sphere(store(i,j),nugget,sill,range)
             elseif(store(i,j).gt.range) then
               C(i,j)=0.0d0
             else
               C(i,j)=sill
            endif
            if(i.ne.j) C(j,i)=C(i,j)
30       continue
35    continue

c   Compute Inverse of Covariance Matrix                              c
      call choldc(C,n)

c  Begin Loop to build Interpolated Map                               c
      do 85 i=nrows,1,-1
            x1=dble(float(i))-0.5d0
         do 80 j=1,ncols
            y1=dble(float(j))-0.5d0

c   Build Variance Vector                                             c
            do 40 k=1,n
              dist=euclid(x1,y1,u(k,1),u(k,2))
              if( (dist.gt.0.0d0).and.(dist.le.range) ) then
                 sigma(k)=sill-sphere(dist,nugget,sill,range)
               elseif(dist.gt.range) then
                 sigma(k)=0.0d0
```

```
               else
                 sigma(k)=sill
              endif
              if(dist.gt.bndwdt) sigma(k)=0.0d0
40          continue

c   Build Optimal Weights Vector                                       c
            do 55 p=1,n
               sum=0.0d0
               do 50 k=1,p
                  sum=sum+C(p,k)*sigma(k)
50             continue
               w1(p)=sum
55          continue

            do 65 p=1,n
               sum=0.0d0
               do 60 k=p,n
                  sum=sum+C(p,k)*w1(k)
60             continue
               weight(p)=sum
65          continue

c  Finally, compute the interpolated value by Generalized Kriging     c
            read(11,*) elev

            sum=0.0
            do 70 k=1,n
               sum=sum+weight(k)*u(k,3)
70          continue

            y=regres(elev)+sum

            write(12,*) y

80       continue
85    continue


      stop
999   print*,'Error reading "u.dat".  Program terminates.             '
      stop
      end

      subroutine choldc(a,n)
c  performs cholesky decomposition of n by n matrix a, then computes  c
c  the inverse of the lower triangular part, and stores the complete  c
c  lower triangular matrix back in a.

      implicit none
      double precision a(50,50),p(50),sum
      integer n,i,j,k

c  Subroutine to decompose a symmetric positive definite matrix       c
c  was taken from Numerical Recipes.  This subroutine also inverts    c
c  the matrix.                                                        c

      do 30 i=1,n
            do 20 j=i,n
                  sum=a(i,j)
                  do 10 k=i-1,1,-1
                        sum=sum-a(i,k)*a(j,k)
10                continue
```

```
                        if(i.eq.j) then
                              if(sum.le.0.0d0) then
                                    print*
                                    print*,'Covariance Matrix not Positive'
                                    print*,'Definite. Program stops.'
                                    stop
                              endif
                              p(i)=sqrt(sum)
                         else
                              a(j,i)=sum/p(i)
                        endif
20          continue
30    continue

c  This loop computes the inverse of the L matrix                    c
      do 50 i=1,n
            a(i,i)=1.0d0/p(i)
            do 45 j=i+1,n
                  sum=0.0d0
                  do 40 k=i,j-1
                        sum=sum-a(j,k)*a(k,i)
40                continue
                  a(j,i)=sum/p(j)
45          continue
50    continue

c  For this program, store the rest of the inverse back in a         c
      do 65 i=1,n-1
         do 60 j=i+1,n
            a(i,j)=a(j,i)
60       continue
65    continue

      return
      end
```