Spatial Interpolation of Lead Concentration and Rainfall Intensity Using R

Farheen Bano[#1], Surya Durbha[*2]

[#]*Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay, 400076, Mumbai, India*

[1]farheenbano94@gmail.com

[*]*Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay, 400076, Mumbai, India*

[2]durbha.courses@gmail.com

I. INTRODUCTION

Spatial and spatio-temporal distributions of both physical and socioeconomic phenomena can be approximated by functions depending on location in a multi-dimensional space, as multivariate scalar, vector, or tensor fields. Typical examples are elevations, climatic phenomena, soil properties, population densities, fluxes of matter, etc. While most of these phenomena are characterised by measured or digitised point data, often irregularly distributed in space and time, visualisation, analysis, and modelling within a GIS are usually based on a raster representation.

Spatial interpolation is the estimation of an unknown attribute values at unsampled points from measurements made at sampled points. It is based on the principle of spatial autocorrelation or spatial dependence, which measures the degree of dependence between near and distant objects. With increasing applications of spatial interpolation, many spatial interpolation models have been developed.

In this paper, studies are carried out using three different interpolation techniques named as inverse distance weighing (IDW), Trend surface interpolation (TSI), and Kriging. Inverse distance weighing (IDW) interpolator is an automatic and relatively easy techniques, as it requires very few parameter from the operator such as it needs only neighbourhood and exponential parameters. Trend surface interpolation (TSI) is used to fitting a statistical model, a trend surface, through the measured point. Kriging is similar to inverse distance weighing (IDW) but the difference is that in this case, weight is based both on the distance between measured point and predicted location but also on overall spatial arrangement of points.

II. METHODOLOGY

A. *Inverse Distance Weighting Method*

The Inverse Distance Weighing (IDW) interpolator is an automatic and relatively easy technique, as it requires very few parameters from the operator, such as search neighbourhood parameters, exponent and eventually smoothing factor, from the operator (Hessl et al., 2007). It is particularly suitable for narrow datasets, where other fitting techniques may be affected by errors (Tomeczak, 2003). The process is highly flexible and allows estimating dataset with trend or anisotropy, in search neighbourhood shaping. Anyhow interpolator's output may be affected by ―bull's eyes or terraces (Burrough and McDonnel, 1988; Liu, 1999). IDW directly implements the assumption that a value of an attribute at an unsampled location is a weighted average of known data points within a local neighbourhood surrounding the unsampled one (Mitas and Mitasova, 1999), as the following formula:

$$Z_j = \frac{\sum_{i=1}^{n} \frac{Z_i}{(h_{ij}+\delta)^{\beta}}}{\sum_{i=1}^{n} \frac{1}{(h_{ij}+\delta)^{\beta}}} \qquad (1)$$

where Z_j is the value at an unsampled location, Z_i are the known values, β is the weight and δ is a smoothing parameter. The separation distancehij between a known and unknown point is measured with is euclidean distance:

$$h_{ij} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \qquad (2)$$

where Δx and Δy are the distances between the unknown point j and the sampled one i according to reference axes.

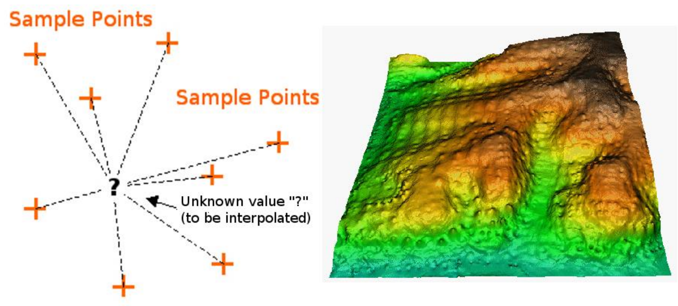

Fig 1. IDW Interpolation; Courtesy: QGIS

The Inverse Distance Weighting (IDW) algorithm effectively is a moving average interpolator that is usually applied to highly variable data. For certain data types it is possible to return to the collection site and record a new value that is statistically different from the original reading but within the general trend for the area.

The interpolated surface, estimated using a moving average technique, is less than the local maximum value and greater than the local minimum value.

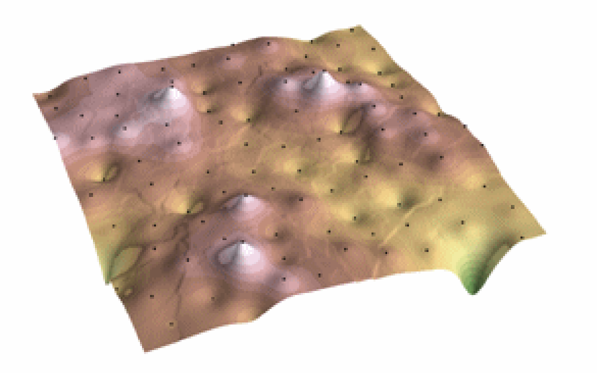

Fig 2. IDW Interpolated Surface; Courtesy:ESRI

IDW interpolation explicitly implements the assumption that things that are close to one another are more alike than those that are farther apart. To predict a value for any unmeasured location, IDW will use the measured values surrounding the prediction location. Those measured values closest to the prediction location will have more influence on the predicted value than those farther away. Thus, IDW assumes that each measured point has a local influence that diminishes with distance. The IDW function should be used when the set of points is dense enough to capture the extent of local surface variation needed for analysis. IDW determines cell values using a linear-weighted combination set of sample points. It weights the points closer to the prediction location greater than those farther away, hence the name inverse distance weighted.

The IDW technique calculates a value for each grid node by examining surrounding data points that lie within a user-defined search radius. Some or all of the data points can be used in the interpolation process. The node value is calculated by averaging the weighted sum of all the points. Data points that lie progressively farther from the node influence the computed value far less than those lying closer to the node.

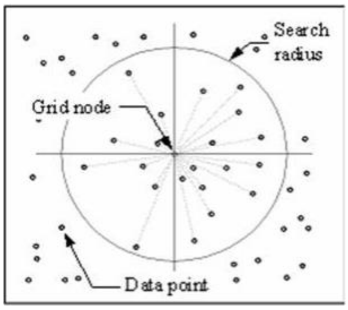

Fig 3. Illustration of IDW

A radius is generated around each grid node from which data points are selected to be used in the calculation. Options to control the use of IDW include power, search radius, fixed search radius, variable search radius and barrier.

Note: The optimal power (p) value is determined by minimizing the root mean square prediction error (RMSPE).

The advantages of IDW interpolation technique:

- Can estimate extreme changes in terrain such as: Cliffs, Fault Lines.
- Dense evenly space points are well interpolated (flat areas with cliffs).
- Can increase or decrease amount of sample points to influence cell values.

The disadvantages of IDW interpolation technique:

- Cannot estimate above maximum or below minimum values.
- Not very good for peaks or mountainous areas.

B. Trend Surface

Trend surface analysis is a mathematical technique for separating a mapable variable into its regional components and local fluctuations.

This approach is aimed to model the overall distribution of properties throughout space. It will sketch the global trend of distribution as a simplified surface.

Trend surface modelling can be applied in the following geographic information context:

- Information can be either in image (raster format) or object (vector format) form
- Properties are estimated at sampled locations, described as a set of geographical locations
- Only spatially continuous distributions can be modelled
- Properties must be quantitative, measured at cardinal level

The principle of a trend surface model is a regression function that estimates the property value Pi at any location, based on the Xi,Yi coordinates of this location. The general function is:

$$Pi = f(Xi, Yi)$$

With:

Pi : property value at location i

Xi , Yi : coordinate values at location i

f : regression function

(3)

A trend surface model is a particular case of a bivariate regression model with two independent variables, the coordinates X and Y and a dependent variable, the thematic variable P to be modelled. One can selecte a linear regression function (first order) or, if the spatial distribution is more complex, a polynomial function (2nd, 3rd, …, or nth order).

The modelled surface will correspond to respectively a flat oriented plane or a curved surface with an increasing number of curvatures.

In order to illustrate the principles of trend surface modelling, let's take a phenomenon with a very obvious and observable spatial distribution: altitude. Let us suppose that we are starting our process of spatial distribution description with a sample of nn data point measurements, irregularly distributed throughout the study area to be described. One can identify 3 stages that are common to most modelling methods:

- In the first step we select the most significant polynomial regression function that best explain the distribution of sample values. This is obtained by computing the F ratio that expresses the proportion of the total variance taken into account by the regression function. As this modelling approach is aimed to sketch the spatial distribution of properties with a continuous surface, it is recommended to limit the order of the regression function up to the fifth order. Such a surface requires 21 coefficients to be modelled. If the F ratio is not yet significant, this would mean that the distribution pattern of properties is simply too complex to be summarised with a surface function.
- Once the regression model has been "calibrated" (i.e. estimation of function coefficient values and selection of the most appropriate function order), the regression function should be then applied to an independent set of sample points for validation purpose.
- Finally, the selected regression function describes the considered trend surface that models the spatial distribution of properties. This function can then be used to estimate the property value at any location Xi,Yi within the study area.

The following figure illustrates the real distribution of altitude values within a study area as well as different trend surfaces modelling this distribution.

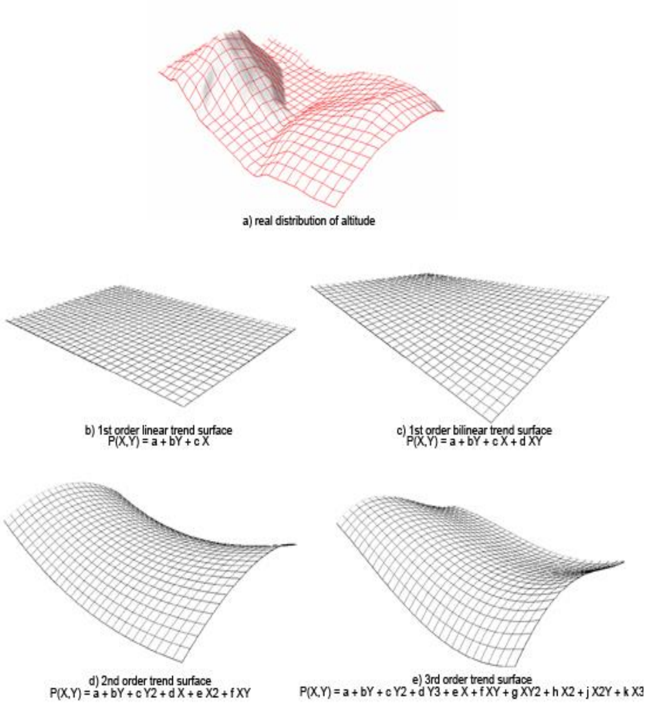

Fig 4. Trend Surface Types

C. Kriging

Kriging is one of several methods that use a limited set of sampled data points to estimate the value of a variable over a continuous spatial field. An example of a value that varies across a random spatial field might be average monthly ozone concentrations over a city, or the availability of healthy foods across neighborhoods. It differs from simpler methods, such as Inverse Distance Weighted Interpolation, Linear Regression, or Gaussian decays in that it uses the spatial correlation between sampled points to interpolate the values in the spatial field: the interpolation is based on the spatial arrangement of the empirical observations, rather than on a presumed model of spatial distribution. Kriging also generates estimates of the uncertainty surrounding each interpolated value.

In a general sense, the kriging weights are calculated such that points nearby to the location of interest are given more weight than those farther away. Clustering of points is also taken into account, so that clusters of points are weighted less heavily (in effect, they contain less information than single points). This helps to reduce bias in the predictions. The kriging predictor is an "optimal linear predictor" and an exact interpolator, meaning that each interpolated value is calculated to minimize the prediction error for that point. The value that is generated from the kriging process for any actually sampled location will be equal to the observed value at this point, and all the interpolated values will be the Best Linear Unbiased Predictors (BLUPs).

Kriging will in general not be more effective than simpler methods of interpolation if there is little spatial autocorrelation among the sampled data points (that is, if the values do not co-vary in space). If there is at least moderate spatial autocorrelation, however, kriging can be a helpful method to preserve spatial variability that would be lost using a simpler method.

Kriging can be understood as a two-step process:

- The spatial covariance structure of the sampled points is determined by fitting a variogram; and
- Weights derived from this covariance structure are used to interpolate values for unsampled points or blocks across the spatial field.

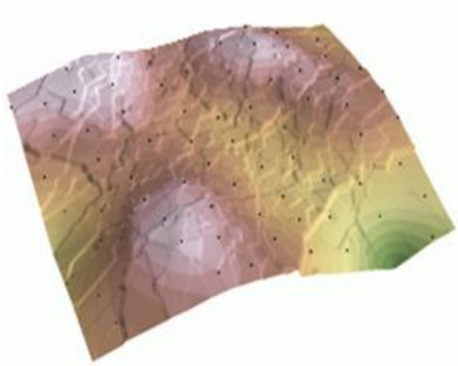

Fig 5. Kriging Surface; Courtesy:ESRI

The kriging model described by Cressie (1993) and Krige (1951) has an approximation function that is composed of a trend term and an autocorrelation term. That is,

$$\hat{f}(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}), \quad (4)$$

where μ(x) is the overall trend and Z(x) is the autocorrelation term. Z(x) is treated as the realization of a mean-zero stochastic process with a covariance structure given by $\mathrm{Cov}(Z(\mathbf{x})) = \sigma^2\mathbf{R}$, where R is an n × n matrix whose (i, j)th element is the correlation function $R(x^i, x^j)$ between any two of the sampled observations x^i and x^j. Ordinary kriging assumes a scalar trend $x=\mu_0$, whereas universal kriging uses a parametric trend term.

The kriging method will be more successful when spatially correlated distance or directional bias is present in the data. It is widely applied in soil science and geology.

C.1. Semivariogram

The semivariogram $\gamma(h)$ was first defined by Matheron (1963) as half the average squared difference between points (s_1 and s_2) separated at distance h. Formally

$$\gamma(h) = \frac{1}{2V}\iiint_V [f(M+h) - f(M)]^2 dV, \quad (5)$$

where M is a point in the geometric field V, and f(M) is the value at that point. For example, suppose we are interested in iron content in soil samples in some region or ield V. M(f) would be the content (e.g., in mg iron per kg soil) of iron at some location M, where M has coordinates of latitude, longitude, and depth. The triple integral is over 3 dimensions. h is the separation distance (e.g., in m or km) of interest. To obtain the semivariogram for a given $\gamma(h)$, all pairs of points at that exact distance would be sampled. In practice it is impossible to sample everywhere, so the empirical variogram is used instead.

C.2. Variogram

The variogram is defined as the variance of the difference between field values at two locations (s_1 and s_2, note change of notation from M to s and f to Z) across realizations of the field (Cressie 1993):

$$2\gamma(\mathbf{s}_1, \mathbf{s}_2) = \mathrm{var}(Z(\mathbf{s}_1) - Z(\mathbf{s}_2)) = E\left[((Z(\mathbf{s}_1) - \mu(\mathbf{s}_1)) - (Z(\mathbf{s}_2) - \mu(\mathbf{s}_2)))^2\right] \quad (6)$$

or in other words is twice the semivariogram.

The semivariogram is a plot of points, which due to spatial autocorrelation, tend to increase in semivariance (*y*-axis) with increasing distance (Fig. 5.8). When we fit a function to those points, we have a model for the rate of decay of spatial autocorrelation's strength and an estimation for its reach or range. Fitting a model for a semivariogram and interpreting its shape is one of the most important ways that the mapper interacts with this spatial model (Oliver and Webster, 2014). Where the model intersects, the *y*-axis is called the nugget. This represents the variance that is expected from repeatedly measuring the same point. It can largely be thought of as representing measurement error, but can also be caused by microscale effects. The model graph should then follow the points with increasing semivariance with increasing distance. Because there is theoretically some limit to the variance a variable can have, the model graph will likely level off to form what is called the sill.

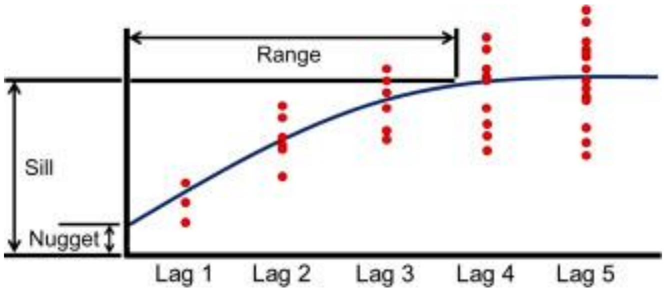

Fig 6. Schematic of a conceptual semivariogram. The red (gray in print versions) dots represent the variety of semivariances observed at each of the lag distances. The blue (black in print versions) curving line is the fitted model, summarizing the trend in semivariance with distance. Important components about the shape of the model include where it intersects the *y*-axis (nugget), where it approaches being level (sill), and the distance at which that leveling occurs (range). The ratio between the nugget and the sill indicates the strength of spatial autocorrelation and the range indicates the reach of spatial autocorrelation for the variable of interest.

By examining the geometry of the fitted model, we can identify two key characteristics about the observed spatial autocorrelation. First the distance (*x*-axis) at which the model reaches the sill is considered the range for the influence of spatial autocorrelation. Second the difference between the sill and the nugget is called the partial sill, which represents the strength of the observed spatial autocorrelation effect. To better describe the relative strength of the spatial autocorrelation, the nugget to sill ratio is regularly used. Interpretation of that ratio can roughly be described as <25% showing strong spatial dependence, 25%–75% showing moderate spatial dependence, and >75% showing weak spatial dependence. However, there are some issues of scale in detecting the spatial autocorrelation. A flat semivariogram (high nugget to sill ratio) could be caused by samples being too far apart to detect spatial autocorrelation occurring more locally.

Based on your semivariogram results, you can select a semivariogram that is spherical, circular, exponential, Gaussian or linear. Alternatively, if you can make an intellectual justification for a mathematical model, then you pick that one.

The empirical variogram cannot be computed at every lag distance h and due to variation in the estimation it is not ensured that it is a valid variogram, as defined above. However some Geostatistical methods such as kriging need valid semivariograms. In applied geostatistics the empirical variograms are thus often approximated by model function ensuring validity (Chiles&Delfiner 1999). Some important models are (Chiles&Delfiner 1999, Cressie 1993):

- *The exponential variogram model*

$$\gamma(h) = (s-n)(1 - \exp(-h/(ra))) + n1_{(0,\infty)}(h) \quad (7)$$

- *The spherical variogram model*

$$\gamma(h) = (s-n)\left(\left(\frac{3h}{2r} - \frac{h^3}{2r^3}\right)1_{(0,r)}(h) + 1_{[r,\infty)}(h)\right) + n1_{(0,\infty)}(h) \tag{8}$$

- *The Gaussian variogram model*

$$\gamma(h) = (s-n)\left(1 - \exp\left(-\frac{h^2}{r^2 a}\right)\right) + n1_{(0,\infty)}(h) \tag{9}$$

The parameter a has different values in different references, due to the ambiguity in the definition of the range.

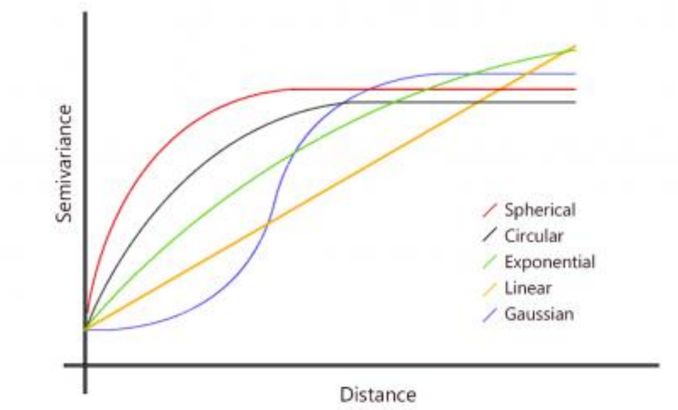

Fig 7. Illustration of different kriging of variogram models

There are several sub-types of kriging, including:

- *Ordinary kriging*, for which the assumption of stationarity (that the mean and variance of the values is constant across the spatial field) must be assumed. This is one of the simplest forms of kriging, but the stationarity assumption is not often met in applications relevant to environmental health, such as air pollution distributions.
- *Universal kriging*, which relaxes the assumption of stationarity by allowing the mean of the values to differ in a deterministic way in different locations (e.g. through some kind of spatial trend), while only the variance is held constant across the entire field. This second-order stationarity (sometimes called "weak stationarity") is often a pertinent assumption with environmental exposures.
- *Block kriging,* which estimates averaged values over gridded "blocks" rather than single points. These blocks often have smaller prediction errors than are seen for individual points.
- *Cokriging*, in which additional observed variables (which are often correlated with each other and the variable of interest) are used to enhance the precision of the interpolation of the variable of interest at each location.
- *Poisson kriging,* for incidence counts and disease rates

Limitations of Kriging Interpolation:

- Kriging assumes that the space being studied is stationary; that is to say, that the joint probability distribution doesn't change throughout the study space.
- It also assumes a property called isotropy; that there is uniformity in every direction.
- If these conditions are difficult to fulfill, the method becomes problematic. However, in universal kriging the stationary requirement is relaxed.
- The accuracy of your model will be limited if the data aren't spatially correlated, if their limited in spread, or if the number of data points are small.

III. Results

A. *Input Data*

The input data consists of lead concentration and rainfall intensity at several locations. These data are provided by several government agencies. These data will be used to predict the lead concentration and rainfall at unmeasured locations using interpolation.

B. *Software- R studio*

R and its libraries were used for the various interpolation methods. gstat and sp package were installed with R for the calculation. gstat package provides a wide range of univariable and multivariable geostatistical modelling, prediction and simulation functions. sp package provides general purpose classes and methods for defining, importing/exporting and visualizing spatial data.

C. *Inverse weighted Distance*

1.) IDW for Rainfall Intensity:

Figure 8, 9, 10 and 11 shows IDW for rainfall data at inverse exponent power of 1,2,5 and 10 respectively. From the figure, we can observed that as IDW power is increasing, areal extent of output value is also increasing. Just take an example from figure, consider yellow dots at power 1 is very small but it is increasing as power increases from 1 to 10, similarly for blue and pink dots.

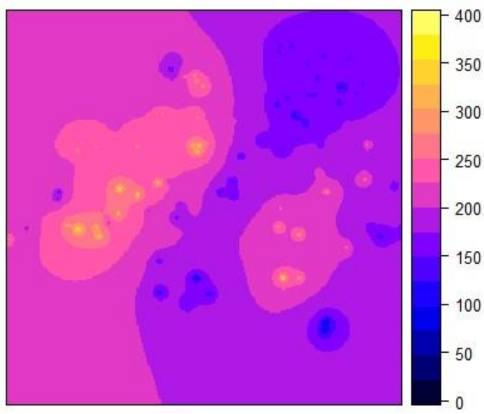

Fig. 8. IDW of Rainfall Intensity with power 1

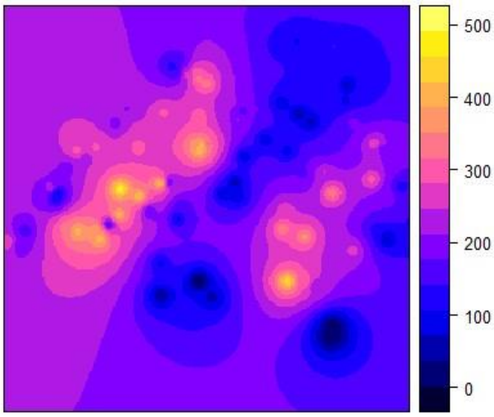

Fig. 9. IDW of Rainfall Intensity with power 2

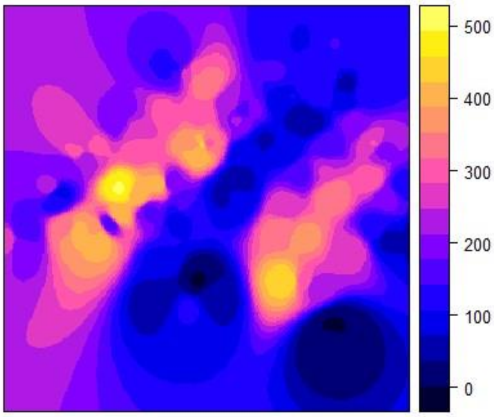

Fig.10. IDW of Rainfall Intensity with power 5

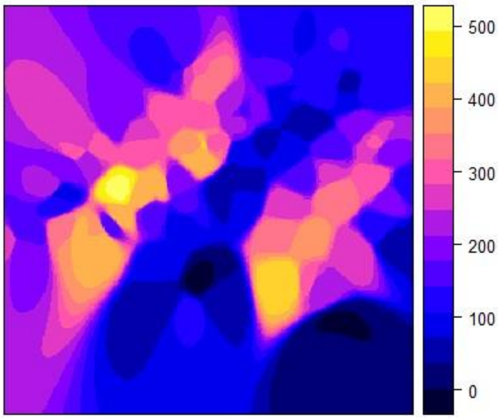

Fig.11. IDW of Rainfall Intensity with power 10

2.) IDW for Lead Concentration:

The same observation can be obtained for lead concentration also, which can be observed in figure 12, 13, 14 and 15 respectively. i.e., yellow, blue and pink dots are increasing with increasing powers. However, the increasing in size of dots is different from rainfall data. It may be due to less data available for lead concentration.

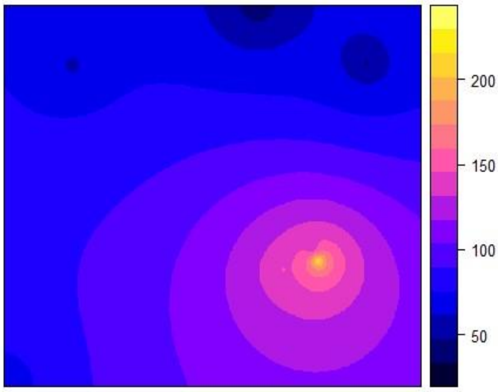

Fig. 12. IDW of Lead Concentration with power 1

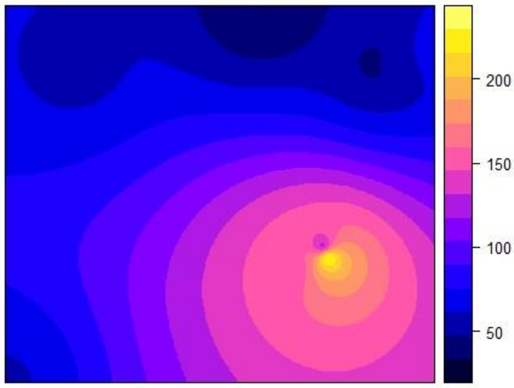

Fig. 13. IDW of Lead Concentration with power 2

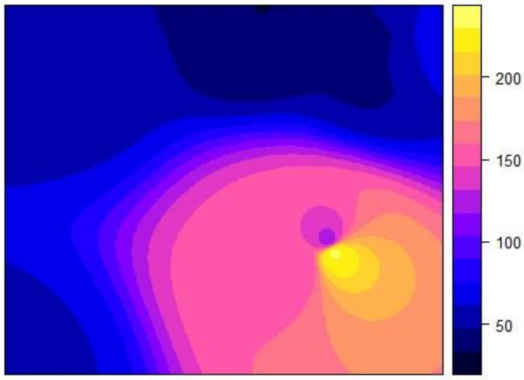

Fig. 14. IDW of Lead Concentration with power 5

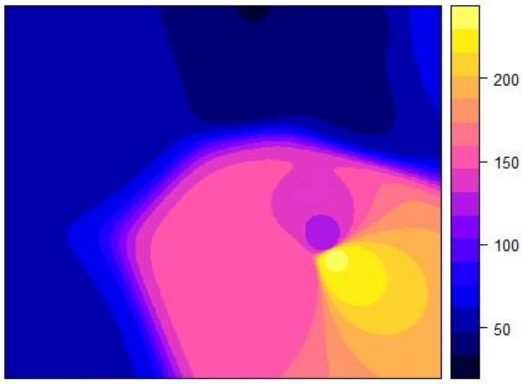

Fig. 15. IDW of Lead Concentration with power 10

D. Trend Surface Interpolation (TSI)

1.) Trend Surface for Rainfall Concentration:

From the figures we can observe that trend is changing smoother in fig 16 and 17(higher order polynomial function) than 1st order polynomial. Also, when we compared with IDW data, it is observed from fig 16 (for rainfall data) that as we move from left to right, the trend is decreasing in that direction (i.e., value decreases from 300 to 100.

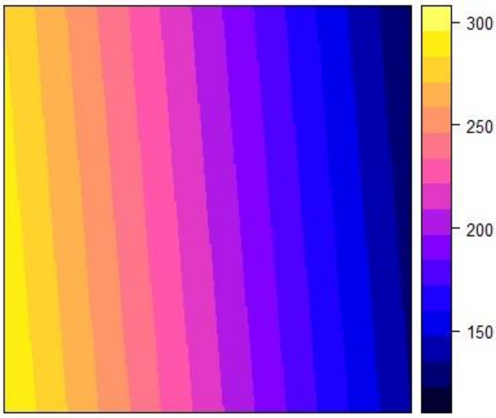

Fig. 16. Trend Surface using 1st Order Polynomial for Rainfall Intensity

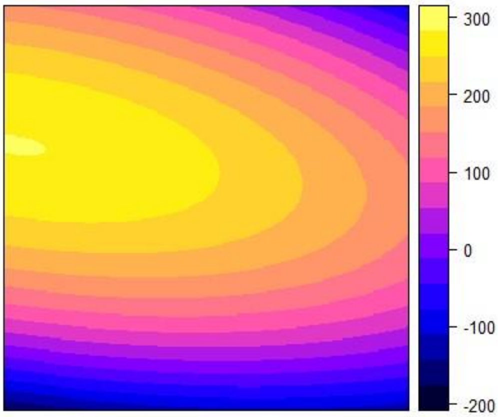

Fig. 17. Trend Surface using 2nd Order Polynomial for Rainfall Intensity

2.) Trend Surface for Trend Concentration:

Similarly, we can observed from fig 18 (i.e., value decreases from 400 to 0). In this way, we can say pattern is similar in both (IDW and trend surface). Also, for lead concentration trend is increasing from left to right in diagonal direction (shown by blue to yellow colour) same as that of IDW plots.

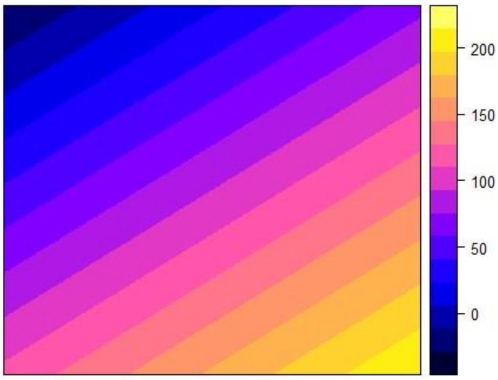

Fig. 18. Trend Surface using 1st Order Polynomial for Lead Concentration

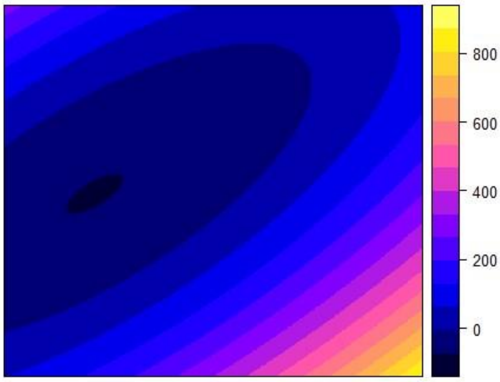

Fig. 19. Trend Surface using 2nd Order Polynomial for Lead Concentration

E. Variogram and Kriging

1.) Variogram and Kriging for Rainfall Intensity:

Fig 20, 23 and 26 shows the semi-variogram of rainfall data for spherical, exponential and gaussian models respectively. Different sill, nugget and range value are taken in order to fit best model.

Kriging surface is produced by using R code. Fig 21, 24 and 27 shows the kriging surfaces by using spherical, exponential and gaussian model respectively. Blue dots showing high rainfall areas and as we move towards white dots showing less rainfall areas.

TABLE I
RAINFALL INTENSITY INPUT VALUES

Model	Sill	Nugget	Range
Spherical	15000	1000	80000
Exponential	15000	1000	80000
Gaussian	15000	1000	80000

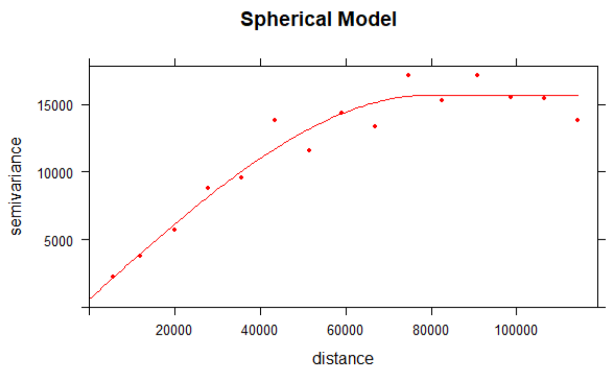

Fig. 20. Spherical Variogram Model for Rainfall Intensity

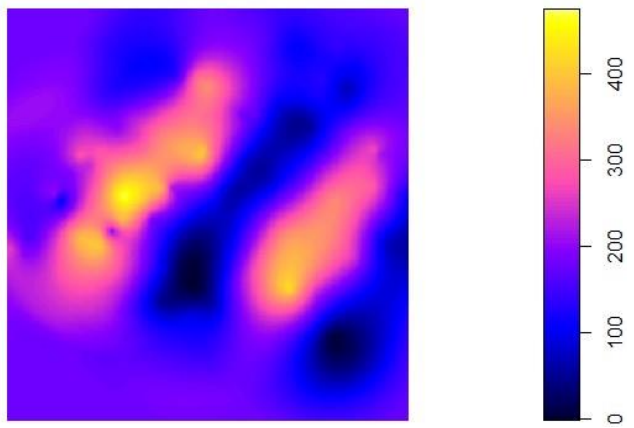

Fig. 21. Kriging Surface of Spherical Model for Rainfall Intensity

```
  model       psill    range
1   Nug    604.7394     0.00
2   Sph  15074.4298 79676.52
```

Fig. 22. Output produced by Spherical Model for Rainfall Intensity showing the values of Nugget, Sill and Range

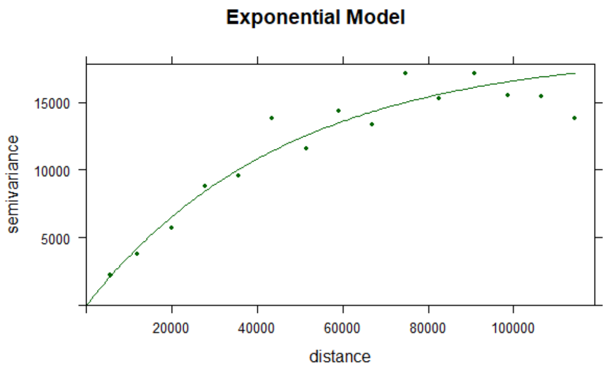

Fig. 23. Exponential Variogram Model for Rainfall Intensity

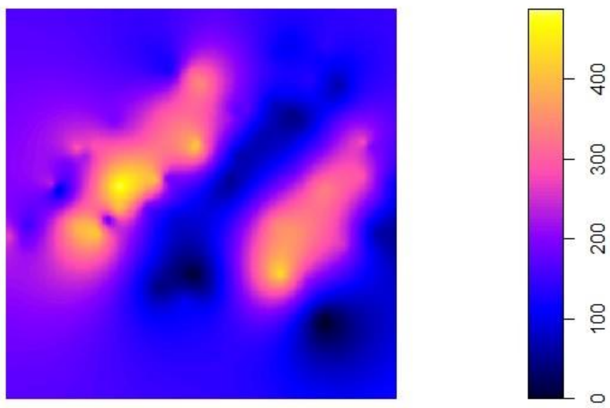

Fig. 24. Kriging Surface of Exponential Model for Rainfall Intensity

```
  model    psill    range
1   Nug     0.00     0.00
2   Exp 18792.71 46767.63
```

Fig. 25. Output produced by Exponential Model for Rainfall Intensity showing the values of Nugget, Sill and Range

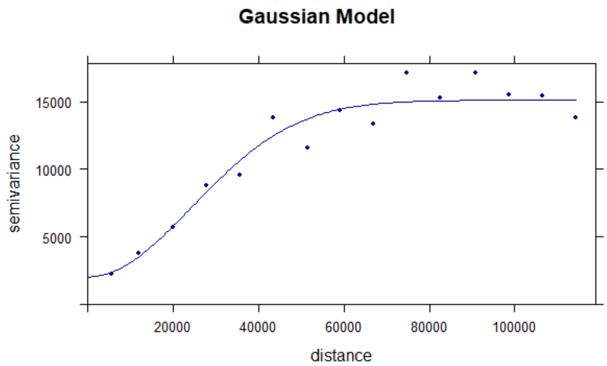

Fig. 26. Gaussian Variogram Model for Rainfall Intensity

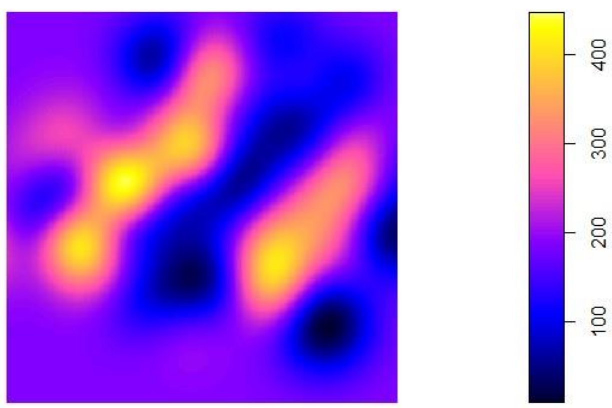

Fig. 27. Kriging Surface of Gaussian Model for Rainfall Intensity

```
  model     psill    range
1   Nug  2024.827     0.00
2   Gau 13075.203 34309.07
```

Fig. 28. Output produced by Gaussian Model for Rainfall Intensity showing the values of Nugget, Sill and Range

2.) Variogram and Kriging for Lead Concentration:

Different values of sill, nugget and range are taken by hit and trial method in order to get best fitted model. Figure 29, 32 and 35 showing the curve using different model like spherical, exponential and gaussian respectively. We took higher and lower value both but lower value gives the best results as we can see from figures line passes through maximum points.

Kriging surface for lead data is generated by using R code using ggplot2. Fig 30,33 and 36 showing the different kriging surface using different models like spherical, exponential and gaussian respectively by putting different nugget, sill and range values. As we move from blue to white color, lead concentration decreases.

TABLE II
LEAD CONCENTRATION INPUT VALUES

Model	Sill	Nugget	Range
Spherical	550	50	0.4
Exponential	500	50	0.25
Gaussian	500	50	0.25

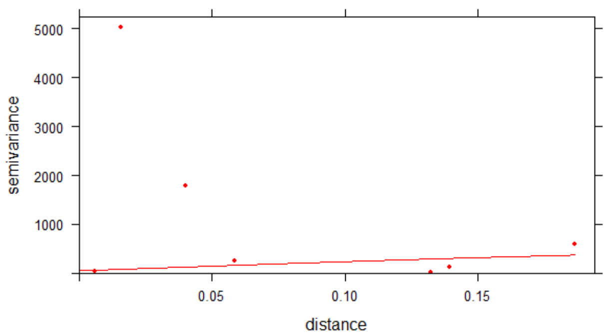

Fig. 29. Spherical Variogram Model for Lead Concentration

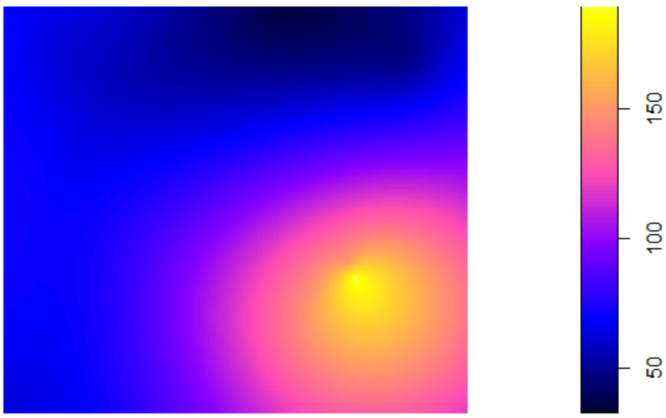

Fig. 30. Kriging Surface of Spherical Model for Lead Concentration

```
  model     psill     range
1   Nug  49.83374 0.0000000
2   Sph 419.56900 0.3259254
```

Fig. 31. Output produced by Spherical Model for Lead Concentration showing the values of Nugget, Sill and Range

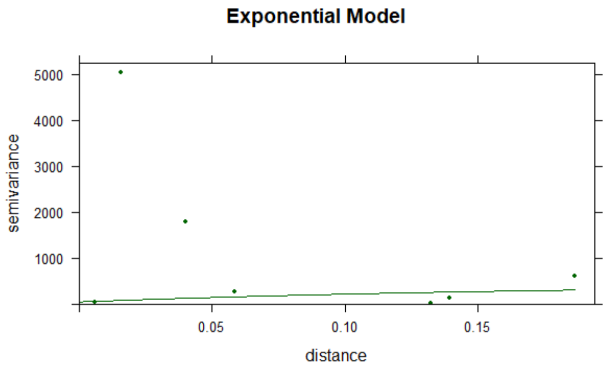

Fig. 32. Exponential Variogram Model for Lead Concentration

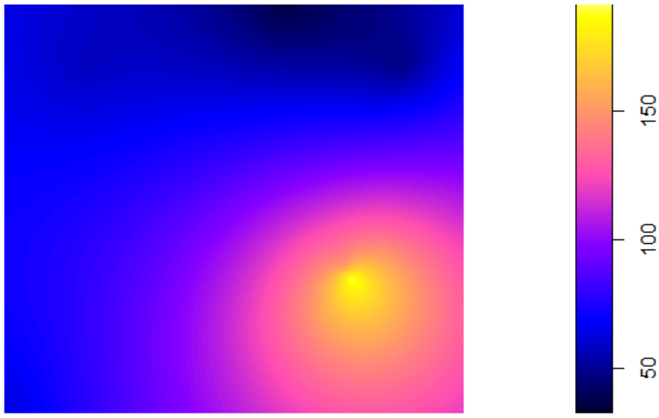

Fig. 33. Kriging Surface of Exponential Model for Lead Concentration

```
  model     psill     range
1   Nug  49.08499 0.0000000
2   Exp 402.26548 0.1831627
```

Fig. 34. Output produced by Exponential Model for Lead Concentration showing the values of Nugget, Sill and Range

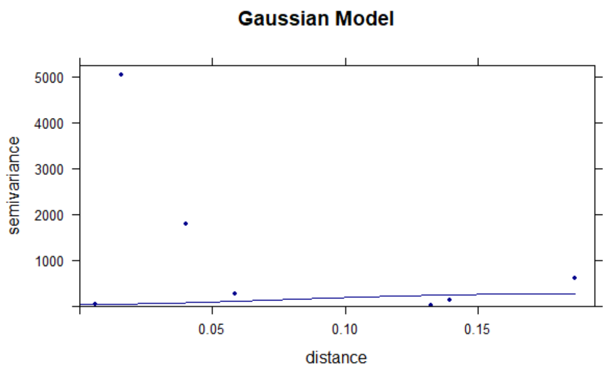

Fig. 35. Gaussian Variogram Model for Lead Concentration

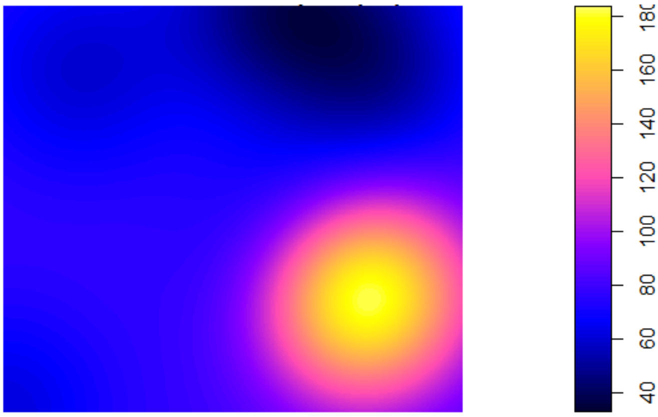

Fig. 36. Kriging Surface of Gaussian Model for Lead Concentration

	model	psill	range
1	Nug	42.6478	0.0000000
2	Gau	247.9271	0.1057505

Fig. 37. Output produced by Gaussian Model for Lead Concentration showing the values of Nugget, Sill and Range

IV. Conclusion

This paper gives the results of rainfall and lead concentration data by comparing three interpolation methods that are used to predict values for unknown points from a known points. It can be observed that all the three methods give a comparable good results. Since different interpolation techniques depend on various parameter so, according to the various techniques i.e., IDW, TSI and kriging, output plots are changing clearly seen from above figures. It can also be observed that since kriging method uses variogram for computing its weight factor so it gives better results for spatial interpolation. We can see in appendices that both rain sample data are more than the lead data samples. Degree of accuracy depend the no. of sample point taken and in case of IDW and TSI, it also depends on exponent power and higher order polynomial used respectively. From this paper, we conclude that all the three methods produce a good results. However, we could not able to identify which model is the best.

Acknowledgment

I would like to express my special thanks of gratitude to my Prof. Surya Durbha, CSRE, IIT Bombay. This paper work would not be completed without the guidance and the kind support of Prof. S. Durbha. I would also like to thank my seniors and batchmate who helped me by giving their suggestions to improve my work.

References

[1] Al-Taee, M. H., Al-Adilli, A., & Sivakugan, N. Comparison of Two Deterministic Interpolation Methods for Predicting Ground Water Level in Baghdad.

[2] Hessl, A., Miller, J., Kernan, J., Keenum, D., McKenzie, D., 2007. Mapping Paleo-Fire Boundaries from Binary Point Data: Comparing Interpolation Methods, The Professional Geographer, 59 (1), pp. 87-104.

[3] Retrive from https://giscourses.cfans.umn.edu/sites/giscourses.cfans.umn.edu/files/interpolation.pdf

[4] Royle, A. G., F. L. Clausen, and P. Frederiksen. "Practical Universal Kriging and Automatic Contouring." Geoprocessing 1: 377–394. 1981.

[5] Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region, Economic Geography, 46(2): 234- 240

[6] zur Erlangung des Doktorgrades der Naturwissenschaften Frankfurt (2007) (D 30) On the Small-Scale Variability of Electrical Soil Properties and its Influence on Geophysical Measurements page no13.

Appendices I

1.) *Rain Intensity data*

"x"	"y"	"z"
101069	21615	200
-23234	24138	426
33933	95354	120
132684	-36715	39
46624	4259	171
66081	-107871	1
-116780	-32596	334
67556	-101183	0
-49564	46539	288
1778	-11490	73
-80712	-24291	407
20697	27464	70
-116742	8555	183
-59304	-22317	174
-103712	14987	225
68402	92319	142
54930	-73510	227
-51974	-57968	99
-123184	-9111	79
34603	-20286	262
-148491	-46357	187
-15935	-81522	58
-127684	-5680	234
-25311	69736	331
96467	24875	282
39426	57582	106
1762	41883	212
81859	40225	184
60382	71103	134
-51393	-80213	39
31917	53089	77
-109680	20655	296
76383	54608	95
53719	59919	119
6322	16312	71
-24666	44159	311
47974	-70233	312
-66815	-11118	405
-93956	-40780	415
-20059	66380	350
44793	45389	45
35888	19752	86
-79712	-6511	493
105188	-40595	65
77743	65746	61
-25141	-70371	0
-42545	77619	133
4779	47445	165
83475	85842	115
81551	-48745	257
-145922	-35296	149
12324	48568	178
-109192	-36075	394
254	-1483	53
-27780	24157	378
-33561	73113	253
61111	73335	126
-67188	23360	300
-87683	-30866	154
-18685	25234	314
-45400	-1310	192
-44038	78742	123
-26794	74190	315
53526	81045	123
-96756	22655	290
-82882	39127	194
-124038	-13542	120
-75197	49033	209
94583	378	298
61816	5509	247
67298	-8888	343
-96147	-34073	383
65456	87838	118
98187	-34040	129
-107305	26173	260
-109172	-34963	394
93505	68200	125
-32168	-89241	108
-8127	-9261	68
-41067	74273	142
-3518	53004	226
-68588	33384	256
58991	59969	118
-52257	-2366	406
-24768	19696	412
-38004	83148	152
42166	95408	100
47728	-39100	359
106209	-11660	186
-158368	-42768	271
63147	23316	155
-37133	-18047	142
115226	-4821	126
33150	-33638	345
-39481	-26928	97
89094	61459	119
33209	90903	113
36429	-70314	444
27284	76417	144
52391	39891	54

2.) *Lead Concentration data*

"x"	"y"	"Pb"
2.699	1.199	56.4
2.972	0.988	229.56
2.965	1.002	129.2
2.908	1.263	34.56
3.031	1.201	52.56
2.933	0.978	146.8
2.623	0.853	59.2
2.902	1.263	32.92
3.085	1.214	68.4
3.025	1.201	40.2

Appendices II

Installed Packages - gstat, sp, ggplot2

1.) IDW and Trend Surface for Rainfall Intensity

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolation_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/rain.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x = seq(-160000, 120000, 1000), y = seq(-160000,
120000, 1000))
plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T
library(gstat)
sampinterp.idw <-
  gstat::idw(formula = z ~ 1,
             locations = spat.samp,
             newdata = spat.grid)

# first argument is a formula.
#   left hand side is the response variable
#   right hand side specifies the trend model variables
#   for idw, r.h.s. MUST be ~1, i.e. constant mean
# second is the spatial data set from which to get the obs.
# third is the set of locations at which to predict

# result is an spatial object with coordinates and two data columns:
#  var1.pred: the predictions for variable 1 (z here)
#  var1.var: the prediction variance.  Not computed for idw

sampinterp.idw@data$var1.pred
# or, if predicting on a grid, can use bubble or spplot to plot the gridded
values
spplot(sampinterp.idw, 'var1.pred')
# default power is 2, can change by specifying idp in call to idw
#power 1
sampinterp1.idw1 <-
  gstat::idw(
    formula = z ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 1
  )

spplot(sampinterp1.idw1, 'var1.pred')


# power 2-- default same as sampinterp.idw,
sampinterp1.idw2 <-
  gstat::idw(
    formula = z ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 2
  )

spplot(sampinterp1.idw2, 'var1.pred')

# power 5,
sampinterp1.idw5 <-
  gstat::idw(
    formula = z ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 5
  )

spplot(sampinterp1.idw5, 'var1.pred')

# power 10,
sampinterp1.idw10 <-
  gstat::idw(
    formula = z ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 10
  )

spplot(sampinterp1.idw10, 'var1.pred')

# compare to previous spplot plot

# Trend surface
sample.lm <- lm(formula=z ~ x + y, data=spat.samp)
sample.lmq <- lm(formula= z ~ x + y + I(x^2) + I(y^2) + I(x*y),
data=spat.samp)
sample.ts <- predict(sample.lm, newdata=spat.grid)
sample.tsq <- predict(sample.lmq, newdata=spat.grid)
spat.grid<- SpatialPixelsDataFrame(spat.grid, data.frame(ts=sample.ts,
tsq=sample.tsq) )
spplot(spat.grid, 'ts')
spplot(spat.grid, 'tsq')
```

2.) IDW and Trend Surface for Lead Concentration

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolation_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/Pbcon.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x=seq(min(sample[,1], na.rm=T),max(sample[,1],
na.rm=T), 0.001), y=seq(min(sample[,2], na.rm=T),max(sample[,2],
na.rm=T), 0.001))
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T
library(gstat)
sampinterp.idw <-
```

```
  gstat::idw(formula = Pb ~ 1,
           locations = spat.samp,
           newdata = spat.grid)

# first argument is a formula.
#   left hand side is the response variable
#   right hand side specifies the trend model variables
#   for idw, r.h.s. MUST be ~1, i.e. constant mean
# second is the spatial data set from which to get the obs.
# third is the set of locations at which to predict

# result is an spatial object with coordinates and two data columns:
#  var1.pred: the predictions for variable 1 (Pb here)
#  var1.var: the prediction variance.  Not computed for idw

sampinterp.idw@data$var1.pred
# or, if predicting on a grid, can use bubble or spplot to plot the gridded
values
spplot(sampinterp.idw, 'var1.pred')
# default power is 2, can change by specifying idp in call to idw
#power 1
sampinterp1.idw1 <-
  gstat::idw(
    formula = Pb ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 1
  )

spplot(sampinterp1.idw1, 'var1.pred')


# power 2-- default same as sampinterp.idw,
sampinterp1.idw2 <-
  gstat::idw(
    formula = Pb ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 2
  )

spplot(sampinterp1.idw2, 'var1.pred')

# power 5,
sampinterp1.idw5 <-
  gstat::idw(
    formula = Pb ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 5
  )

spplot(sampinterp1.idw5, 'var1.pred')

# power 10,
sampinterp1.idw10 <-
  gstat::idw(
    formula = Pb ~ 1,
    locations = spat.samp,
    newdata = spat.grid,
    idp = 10
  )

spplot(sampinterp1.idw10, 'var1.pred')

# compare to previous spplot plot

# Trend surface
sample.lm <- lm(formula=Pb ~ x + y, data=spat.samp)
sample.lmq <- lm(formula= Pb ~ x + y + I(x^2) + I(y^2) + I(x*y),
data=spat.samp)
sample.ts <- predict(sample.lm, newdata=spat.grid)
sample.tsq <- predict(sample.lmq, newdata=spat.grid)
spat.grid<- SpatialPixelsDataFrame(spat.grid, data.frame(ts=sample.ts,
tsq=sample.tsq) )
spplot(spat.grid, 'ts')
spplot(spat.grid, 'tsq')
```

3.) Variogram and Kriging Surface for Rainfall Intensity

#Spherical Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/rain.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x = seq(-160000, 120000, 1000), y = seq(-160000,
120000, 1000))
plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(z~1, locations=~x+y, data=sample)
variog1 <- variogram(z~1, locations=~x+y, Cressie=TRUE,
                data=sample)
variog2 <- variogram(z~1, locations=~x+y,
                boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=NA, model="Sph", nugget=NA, range=NA)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="red" )


RainOKrig= krige(z~1, spat.samp, spat.grid, model=fit.variog)

plot(RainOKrig)
```

#Exponential Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/rain.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')
```

```
# construct a grid of locations to predict at
grid <- expand.grid(x = seq(-160000, 120000, 1000), y = seq(-160000,
120000, 1000))
plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(z~1, locations=~x+y, data=sample)
variog1 <- variogram(z~1, locations=~x+y, Cressie=TRUE,
                data=sample)
variog2 <- variogram(z~1, locations=~x+y,
                boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=NA, model="Exp", nugget=NA, range=NA)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="red" )


RainOKrig= krige(z~1, spat.samp, spat.grid, model=fit.variog)

plot(RainOKrig)
```

#Gaussian Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/rain.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x = seq(-160000, 120000, 1000), y = seq(-160000,
120000, 1000))
plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(z~1, locations=~x+y, data=sample)
variog1 <- variogram(z~1, locations=~x+y, Cressie=TRUE,
                data=sample)
variog2 <- variogram(z~1, locations=~x+y,
                boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=NA, model="Gau", nugget=NA, range=NA)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="red" )


RainOKrig= krige(z~1, spat.samp, spat.grid, model=fit.variog)

plot(RainOKrig)
```

4.) Variogram and Kriging Surface for Lead Concentration

#Spherical Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/pbcon.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x=seq(min(sample[,1], na.rm=T),max(sample[,1],
na.rm=T), 0.001), y=seq(min(sample[,2], na.rm=T),max(sample[,2],
na.rm=T), 0.001))
plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(Pb~1, locations=~x+y, data=sample)
variog1 <- variogram(Pb~1, locations=~x+y, Cressie=TRUE,
                data=sample)
variog2 <- variogram(Pb~1, locations=~x+y,
                boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=500, model="Sph", nugget=50, range=0.4)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="red" )


LeadOKrig= krige(Pb~1, spat.samp, spat.grid, model=fit.variog)

plot(LeadOKrig, main="Spherical Surface by Kriging")
```

#Exponential Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/pbcon.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x=seq(min(sample[,1], na.rm=T),max(sample[,1],
na.rm=T), 0.001), y=seq(min(sample[,2], na.rm=T),max(sample[,2],
na.rm=T), 0.001))
plot(grid)
spat.grid <- grid
```

```
# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(Pb~1, locations=~x+y, data=sample)
variog1 <- variogram(Pb~1, locations=~x+y, Cressie=TRUE,
                     data=sample)
variog2 <- variogram(Pb~1, locations=~x+y,
                     boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=500, model="Exp", nugget=50, range=0.25)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="darkgreen" ,main="Exponential
Model")


LeadOKrig= krige(Pb~1, spat.samp, spat.grid, model=fit.variog)

plot(LeadOKrig, main="Exponential Surface by Kriging")
```

#Gaussian Model

```
library(gstat)

library(sp)

sample <-
  read.table(

"C:/Users/Farheen/Downloads/GNR605_Assignment2_R_Spatial_Interpolati
on_2019/GNR605_Assignment2_R_Spatial_Interpolation_2019/pbcon.txt",
    header = T
  )

# and convert to sp object
spat.samp <- sample
coordinates(spat.samp) <- c('x', 'y')

# construct a grid of locations to predict at
grid <- expand.grid(x=seq(min(sample[,1], na.rm=T),max(sample[,1],
na.rm=T), 0.001), y=seq(min(sample[,2], na.rm=T),max(sample[,2],
na.rm=T), 0.001))

plot(grid)
spat.grid <- grid

# convert grid to a SpatialPoints object
coordinates(spat.grid) <- c('x', 'y')

# and tell sp that this is a grid
gridded(spat.grid) <- T

#variogram
variog1<- variogram(Pb~1, locations=~x+y, data=sample)
variog1 <- variogram(Pb~1, locations=~x+y, Cressie=TRUE,
                     data=sample)
variog2 <- variogram(Pb~1, locations=~x+y,
                     boundaries=c(0,1,2,3,4,5,6,7,8,9), data=sample)

model.variog <- vgm(psill=500, model="Gau", nugget=50, range=0.25)
print(fit.variog <- fit.variogram(variog1, model.variog))
plot(variog1, model = fit.variog,pch=20,col="darkblue",main="Gaussian
Model" )

LeadOKrig= krige(Pb~1, spat.samp, spat.grid, model=fit.variog)

plot(LeadOKrig, main="Gaussian Surface by Kriging")
```