

Introduction_to_spatial_statistics__#1

May 4, 2018

0.1 Lezione :: Analisi statistica spaziale di alcune proprietà dei suoli

0.1.1 Laurea magistrale in scienze forestali ed ambientali. Corso di Geografia e Valutazione del suolo

Giuliano Langella glangella@unina.it

Tobler's Law of Geography (1970): "Everything is related to everything else, but near things are more related than distant things".

0.1.2 Step #1: Come si costruisce un variogramma sperimentale

Nota: Il kernel utilizzato per Jupyter è MatLab per cui il codice è in tale linguaggio. Ai fini dello studio degli argomenti trattati è possibile ignorare i comandi MatLab e concentrarsi solo sulla procedura.

Directory di lavoro:

```
In [1]: cd ~/git/seminars/GeogrValutSuolo/
```

Rappresentazione cartografica dei punti di campionamento: crea 2 punti, $P_1(1,1)$ e $P_2(5,4)$:

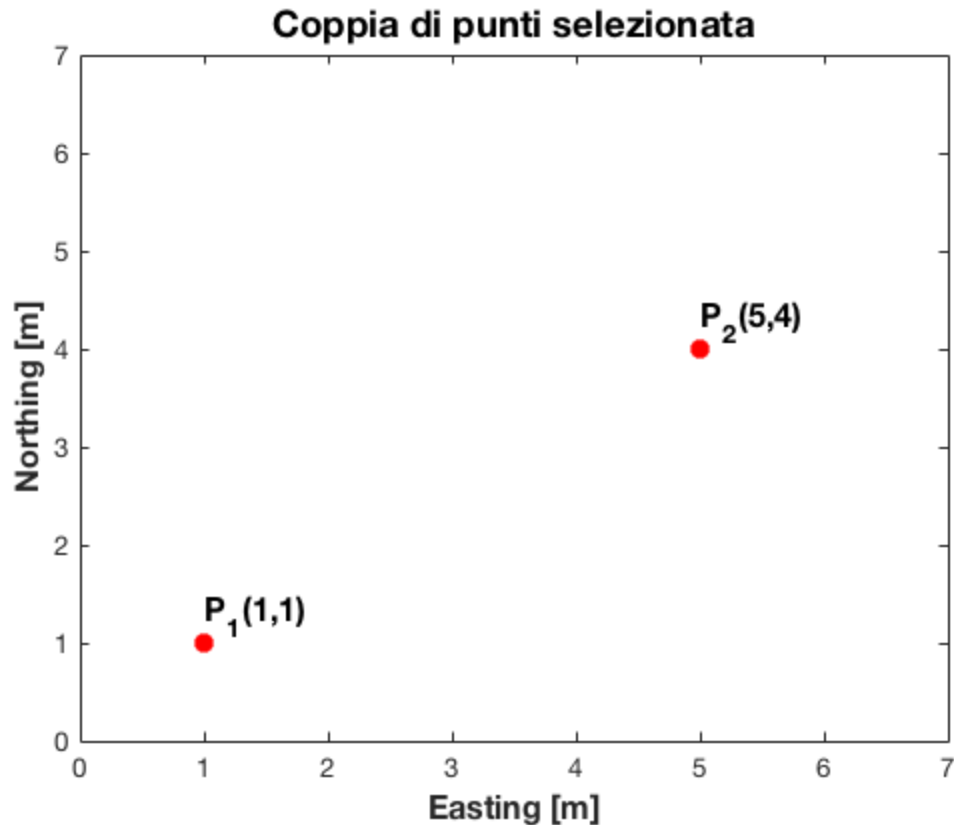
```
In [2]: x=[1,5];
        y=[1,4];
        % P1(1,1)
        P1 = ['P_1(',num2str(x(1))',' ',num2str(y(1))',' '];
        % P2(5,4)
        P2 = ['P_2(',num2str(x(2))',' ',num2str(y(2))',' '];

In [3]: % FIGURE:
        figure
        scatter(x,y,450,'.','r'),axis([0,7,0,7])
        xlabel('Easting [m'],'FontWeight','b','FontSize',12)
```

```

ylabel('Northing [m]', 'FontWeight', 'b', 'FontSize', 12)
title('Coppia di punti selezionata', 'FontWeight', 'b', 'FontSize', 14)
text(x(1), y(1), P1, 'HorizontalAlignment', 'left', ...
     'VerticalAlignment', 'bottom', 'FontWeight', 'b', 'FontSize', 12)
text(x(2), y(2), P2, 'HorizontalAlignment', 'left', ...
     'VerticalAlignment', 'bottom', 'FontWeight', 'b', 'FontSize', 12)
box on

```



Crea 100 punti geospaziali di coordinate $P_i(X_i, Y_i)$ casuali. (funzione MatLab RANDI) Crea una vettore di 100 punti (ossia matrice di dimensioni 100x1) con coordinate casuali (randi) ed intere (randi):

X nel range [1,550]

Y nel range [1,700]

La divisione per 100 obbliga i valori tra 0 e 5.50 / 7.00

```

In [4]: X=randi(550,100,1)/100;% con intervallo da 0 a 5.50
        Y=randi(700,100,1)/100;% con intervallo da 0 a 7.00
        fprintf('%14s, %14s\n','Easting [m]','Northing [m]')
        % osserviamo i primi 5 valori dei vettori coordinate X ed Y:
        for ii=1:5 fprintf('%14.3f, %14.3f\n',X(ii),Y(ii)); end

```

```

Easting [m],    Northing [m]
    4.490,         1.140

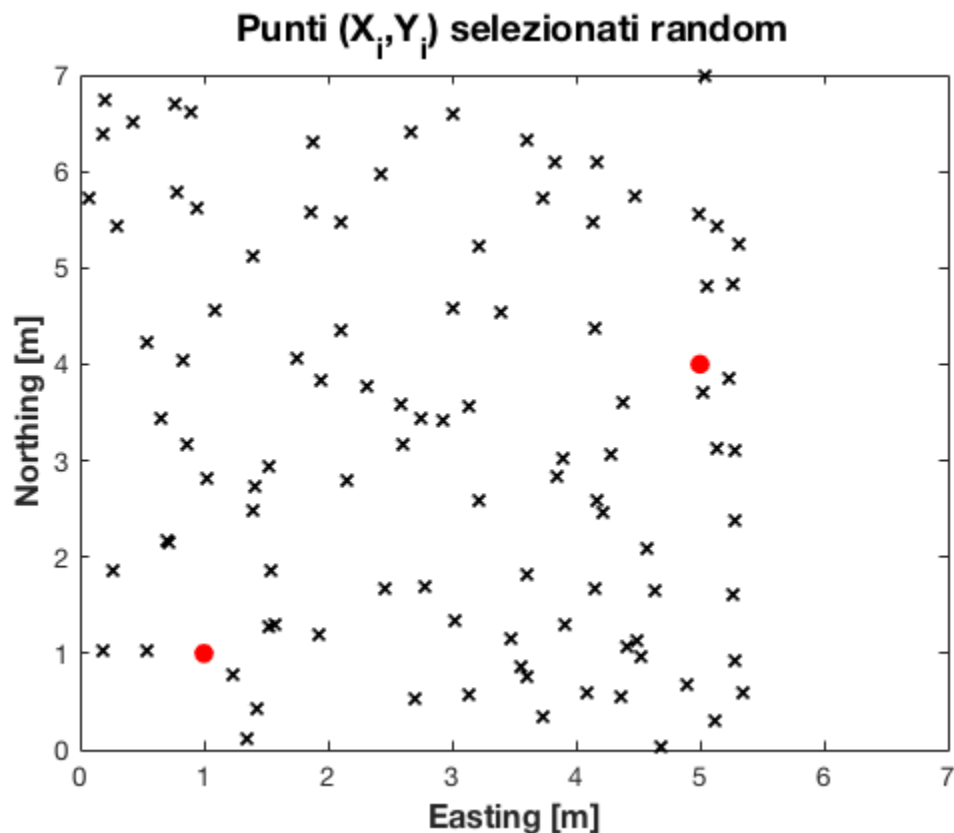
```

4.990,	5.560
0.700,	2.180
5.030,	3.700
3.480,	1.160

Crea una vettore di 100 punti (ossia matrice di dimensioni 100x1)
 con valori casuali di Carbonio Organico $\left[\frac{g}{kg}\right]$
 nell'orizzonte superficiale (es. A_p o profondità fissa 0-40 cm)
 La divisione per 1000 obbliga i valori tra 0.000 e 2.000 $\frac{g}{kg}$

In [5]: `Z=randi(2000,100,1)/1000;`*% con intervallo da 1 a 2.000 g/kg*

In [6]: `figure,hold on`
`scatter(X,Y,07,'x','k'),axis([0,7,0,7])`
`scatter(x,y,450,'.','r'),axis([0,7,0,7])`
`hold off`
`xlabel('Easting [m]','FontWeight','b','FontSize',12)`
`ylabel('Northing [m]','FontWeight','b','FontSize',12)`
`title('Punti (Xi,Yi) selezionati random','FontWeight','b','FontSize',14)`
`box on`



Note sul campionamento: Area/Scala di indagine;
Localizzazione e densità dei punti di campionamento;
Tipologie di campionamento (profili, pit, trivellate, osservazioni, ...);
Scopo del campionamento: osservare l'ubicazione ed i tipi di suoli nell'area di indagine;
Raccolta di campioni di suolo, per ciascun orizzonte, per le analisi fisico-chimiche;
C'è un lavoro preliminare per ottenere dati sui suoli utili alle indagini (geo-)statistiche;

Richiamo delle nozioni cartografiche: Meridiani (Greenwich) e Paralleli (Equatore);
Geoide vs Ellissoide (WGS84);
Sistema di coordinate (Geocentrico, Geografico, Piano);
Trasformazioni tra sistemi;
Datum (WGS84, Roma40, ...):

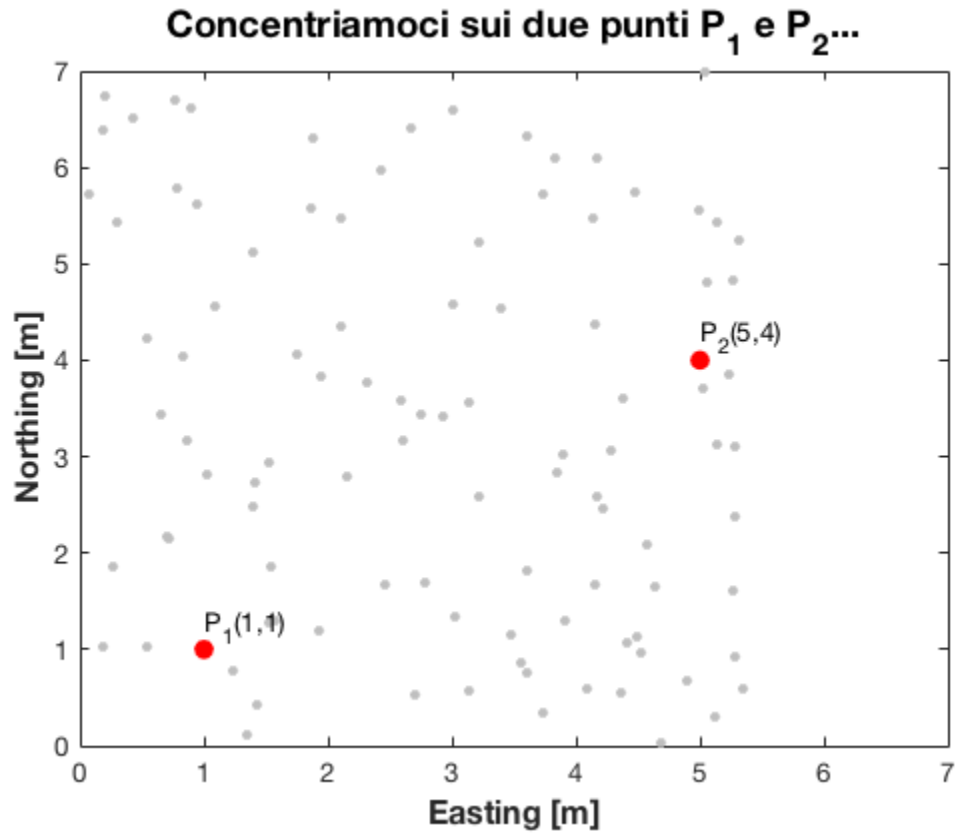
```
<li> scegliere lellissoide, </li>  
<li> eseguirne una materializzazione, </li>  
<li> orientarlo in un punto (verticale corrispondente alla normale), </li>
```

Le analisi statistiche spaziali richiedono sistemi di riferimento piani (proiezioni)!

Sistemi di riferimento: geografico, Globale [EPSG:4326]
piano, Italia Fuso Ovest [EPSG:32632]
piano, Italia Fuso Est [EPSG:32633]

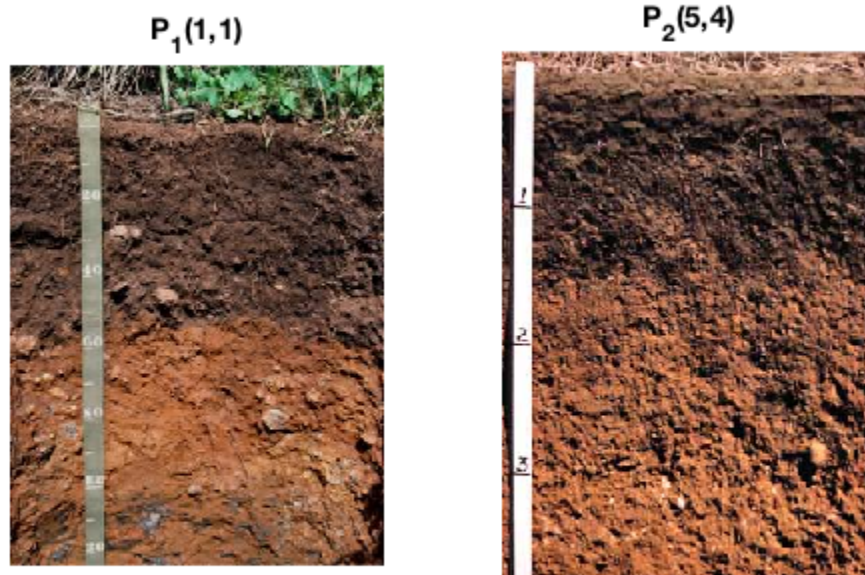
Riprendiamo i due punti a scelti a caso di cui prima, P_1 e P_2 ...

```
In [7]: hold on  
        scatter(x,y,450,'.','r'),axis([0,7,0,7])  
        xlabel('Easting [m]'),ylabel('Northing [m]')  
        scatter(X,Y,16,[.75 .75 .75],'filled'),axis([0,7,0,7])  
        hold off  
        xlabel('Easting [m]','FontWeight','b','FontSize',12)  
        ylabel('Northing [m]','FontWeight','b','FontSize',12)  
        title('Concentriamoci sui due punti P_1 e P_2...',...  
              'FontWeight','b','FontSize',14)  
        text(x(1),y(1),P1,'HorizontalAlignment','left',...  
             'VerticalAlignment','bottom')  
        text(x(2),y(2),P2,'HorizontalAlignment','left',...  
             'VerticalAlignment','bottom')  
        box on
```

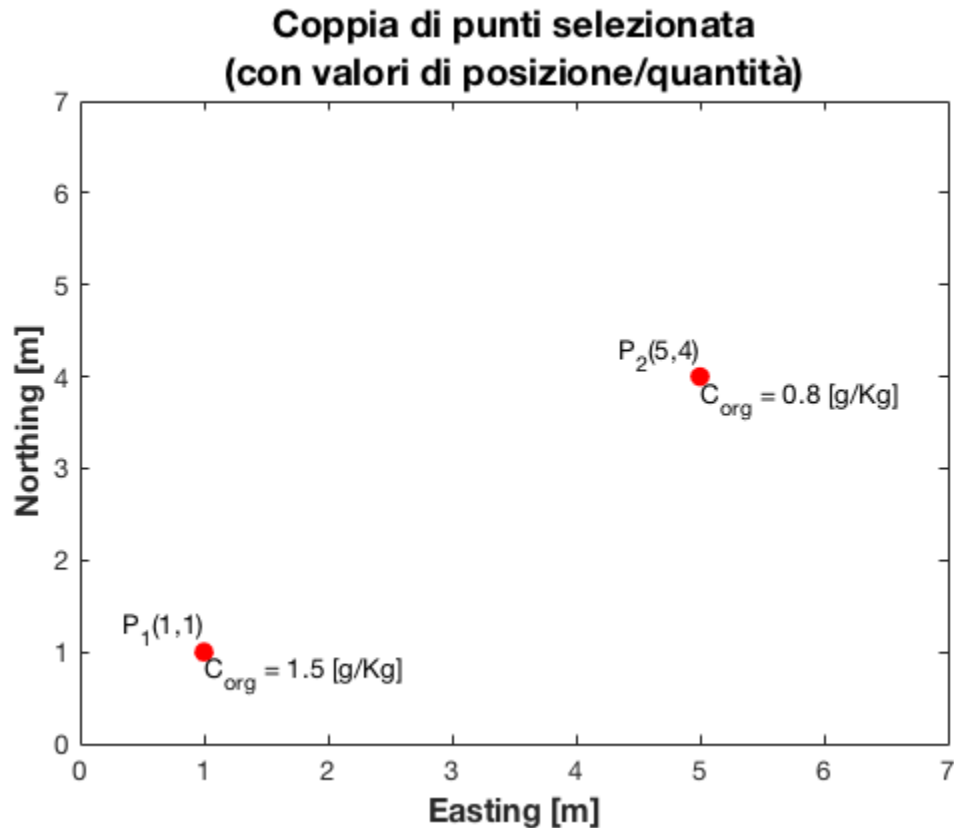


...ciascuno di essi avrà un valore specifico di una proprietà del suolo: ad esempio per il Carbonio Organico (orizz. $A_{(p)}$):

```
In [8]: subplot(121)
        imshow('artwork/P1.jpg'),title(P1)
        subplot(122)
        imshow('artwork/P2.jpg'),title(P2)
```



```
In [9]: z1 = 1.5;% [g/Kg]
        z2 = 0.8;% [g/Kg]
        figure
        scatter(x,y,450,'.','r'),axis([0,7,0,7])
        xlabel('Easting [m'],'FontWeight','b','FontSize',12)
        ylabel('Northing [m'],'FontWeight','b','FontSize',12)
        title({'Coppia di punti selezionata';'(con valori di posizione/quantità)'},...
              'FontWeight','b','FontSize',14)
        text(x(1),y(1),P1,'HorizontalAlignment','right','VerticalAlignment','bottom')
        text(x(2),y(2),P2,'HorizontalAlignment','right','VerticalAlignment','bottom')
        text(x(1),y(1),['C_{org} = ',num2str(z1),' [g/Kg]'],...
              'HorizontalAlignment','left','VerticalAlignment','top')
        text(x(2),y(2),['C_{org} = ',num2str(z2),' [g/Kg]'],...
              'HorizontalAlignment','left','VerticalAlignment','top')
        box on
```



0.1.3 Obiettivo —> calcolare la distanza tra i due punti P_1 e P_2

POSIZIONE :: Distanza geospaziale tra i due punti

QUANTITA' :: Distanza relativa alla variabile (proprietà del suolo | C.O.) misurata nei due punti)

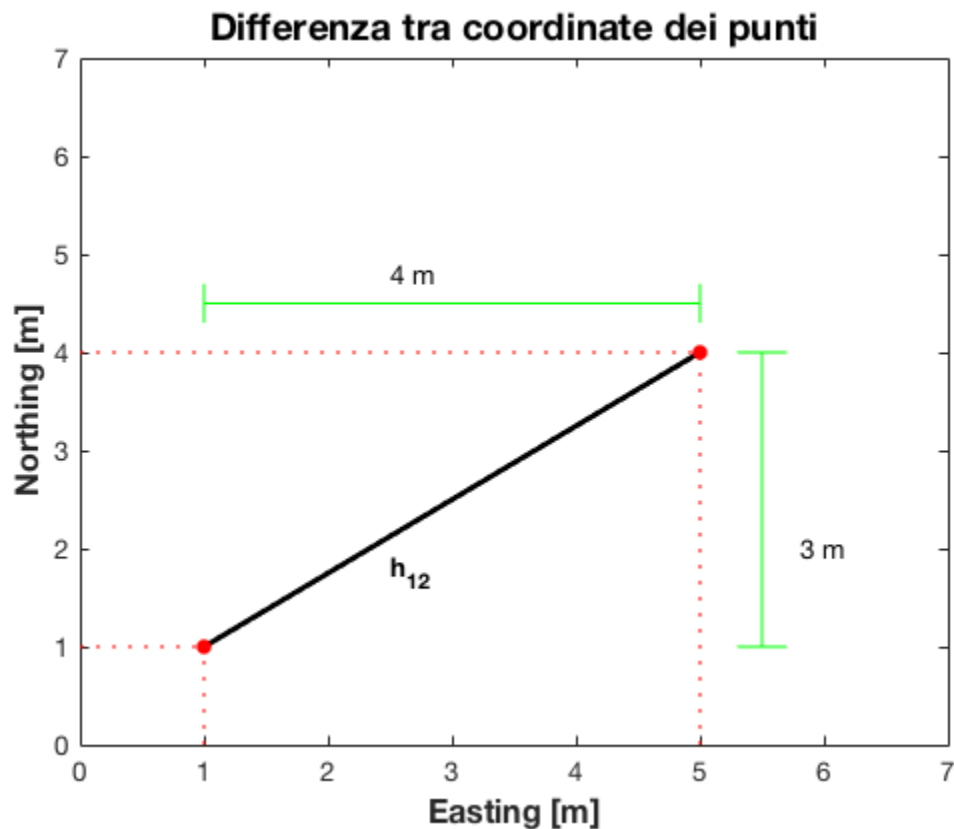
1. Calcoliamo la distanza euclidea tra i due punti geospaziali P_1 e P_2

```
In [10]: hold on
          plot(x,y,'-k','LineWidth',2)
          scatter(x,y,255,'.','r'),axis([0,7,0,7])
          xlabel('Easting [m]'),ylabel('Northing [m]')
          plot([x(1),x(1)],[0,y(1)],'r:')
          plot([x(2),x(2)],[0,y(2)],'r:')
          plot([0,x(1)],[y(1),y(1)],'r:')
          plot([0,x(2)],[y(2),y(2)],'r:')
          plot([x(1),x(2)],[y(2)+0.5,y(2)+0.5],'g')
          plot([1,1],[y(2)+0.5-0.2,y(2)+0.5+0.2],'g')
          plot([x(2),x(2)],[y(2)+0.5-0.2,y(2)+0.5+0.2],'g')
          plot([x(2)+0.5,x(2)+0.5],[y(1),y(2)],'g')
          plot([x(2)+0.5-0.2,x(2)+0.5+0.2],[y(1),y(1)],'g')
```

```

plot([x(2)+0.5-0.2,x(2)+0.5+0.2],[y(2),y(2)], 'g')
hold off
text(x(2)/2,y(2)+0.8,[num2str(x(2)-x(1)),' m'])
text(x(2)+0.8,y(2)/2,[num2str(y(2)-y(1)),' m'])
text(x(2)/2,y(2)/2,['h_{12}'],'HorizontalAlignment','left',...
      'VerticalAlignment','top','FontWeight','b')
xlabel('Easting [m]','FontWeight','b','FontSize',12)
ylabel('Northing [m]','FontWeight','b','FontSize',12)
title('Differenza tra coordinate dei punti','FontWeight','b','FontSize',14)
box on

```



Applichiamo il teorema di Pitagora: $h_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{4^2 + 3^2} = \sqrt{16 + 9} = \sqrt{25} = 5 \text{ [m]}$

2. Varianza: Distanza in C.O. [g/Kg] misurato nei due punti **P₁** e **P₂**

```

In [11]: hold on
plot(x,y,'-k','LineWidth',2)
scatter(x,y,255,'.','r'),axis([0,7,0,7])
xlabel('Easting [m]'),ylabel('Northing [m]')
hold off

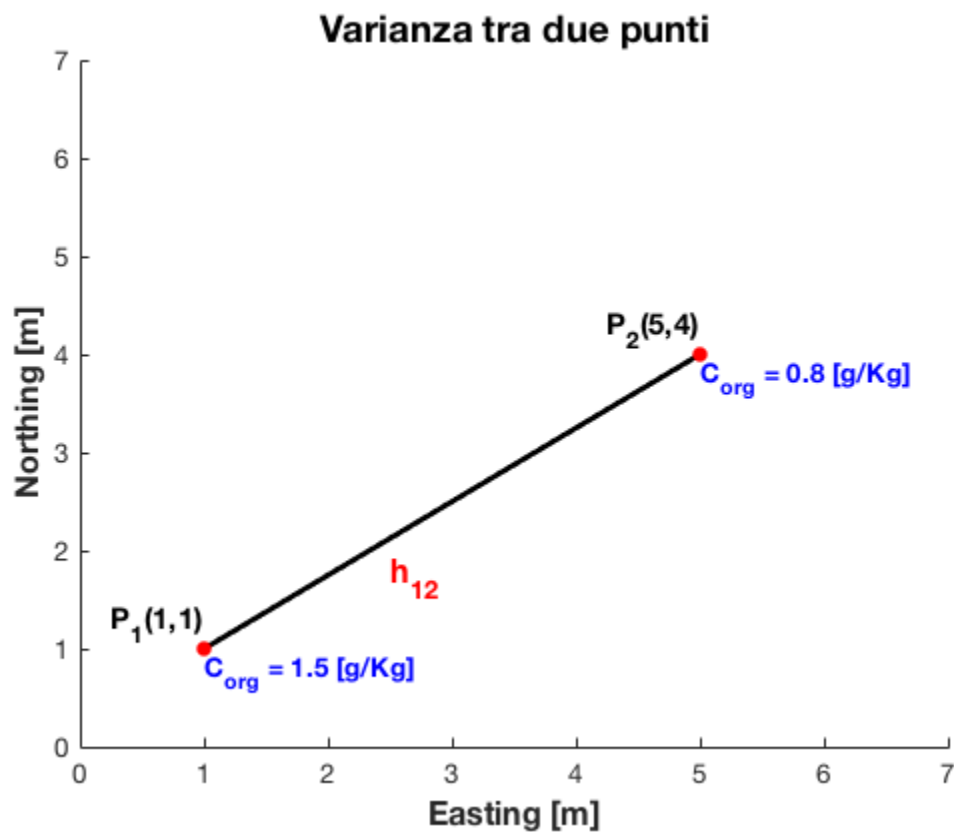
```



```

text(x(2)/2,y(2)/2,['\color{red}h_{12}'],'HorizontalAlignment','left',...
      'VerticalAlignment','top','FontWeight','b','FontSize',12)
text(x(1),y(1),P1,'HorizontalAlignment','right',...
      'VerticalAlignment','bottom','FontWeight','b','FontSize',11)
text(x(2),y(2),P2,'HorizontalAlignment','right',...
      'VerticalAlignment','bottom','FontWeight','b','FontSize',11)
text(x(1),y(1),['C_{org} = ',num2str(z1),' [g/Kg]'],'color','blue', ...
      'HorizontalAlignment','left','VerticalAlignment','top',...
      'FontWeight','b','FontSize',10)
text(x(2),y(2),['C_{org} = ',num2str(z2),' [g/Kg]'],'color','blue', ...
      'HorizontalAlignment','left','VerticalAlignment','top',...
      'FontWeight','b','FontSize',10)
xlabel('Easting [m]','FontWeight','b','FontSize',12)
ylabel('Northing [m]','FontWeight','b','FontSize',12)
title('Varianza tra due punti','FontWeight','b','FontSize',14)

```



$$CO_{P_1} = 1.5 \text{ [g/Kg]} \quad > \quad z(P_1) = z(x_i) \quad CO_{P_2} = 0.8 \text{ [g/Kg]} \quad > \quad z(P_2) = z(x_i + h)$$

Utilizziamo z per indicare il contenuto in C.O. degli orizzonti A_p , ma può essere utilizzata per indicare una qualsiasi proprietà del suolo (e non).

$$\gamma(h_{12}) = (z(x_1 + h_{12}) - z(x_1))^2 = (0.8 - 1.5)^2 = -0.7^2 = 0.49$$

```

In [12]: gamma = (z2 - z1)^2;
          fprintf('gamma = %.2f',gamma)

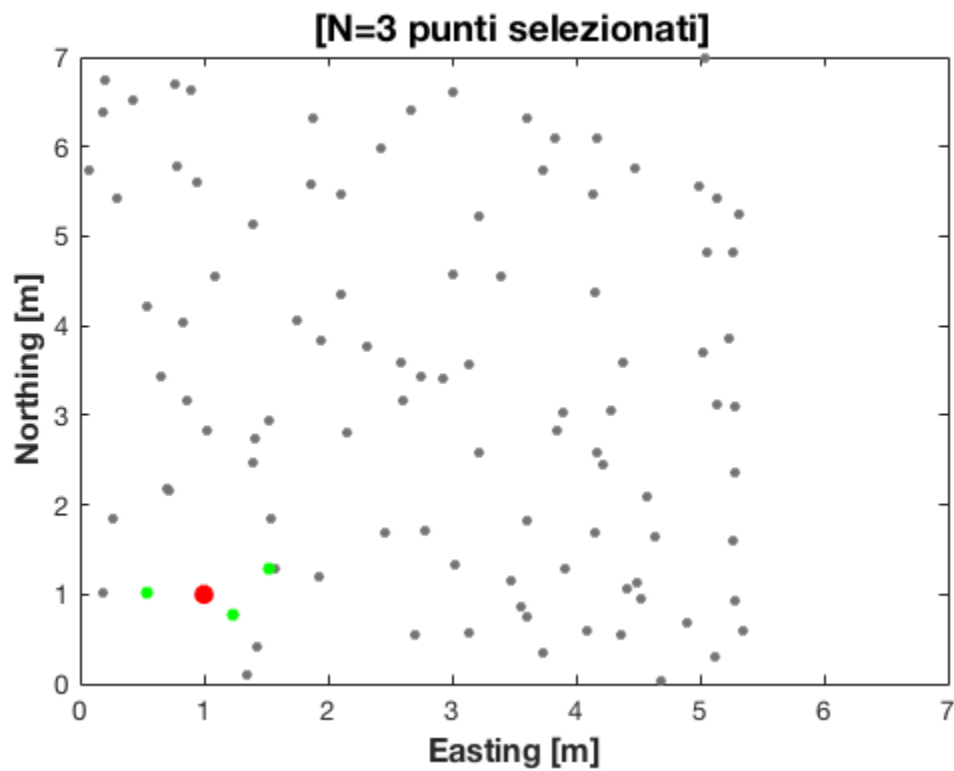
```

gamma = 0.49

0.1.4 Estendiamo il calcolo a più punti contigui, fissando P_1

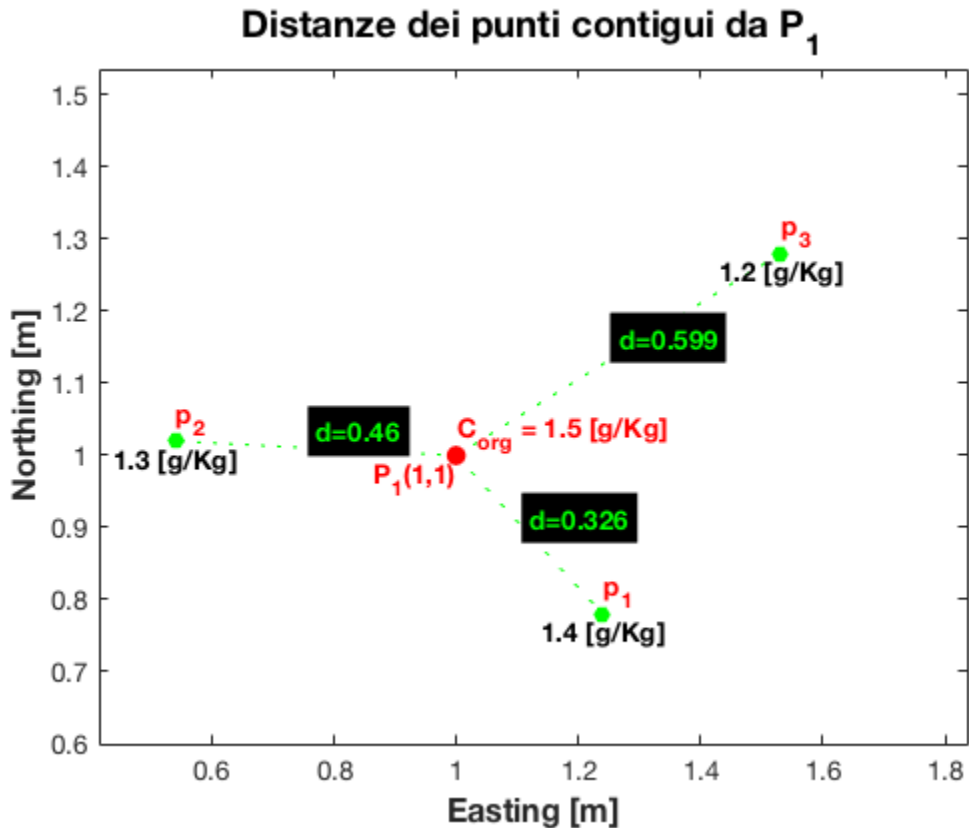
```
In [13]: N = 3;
D = sqrt((X - x(1)).^2 + (Y - y(1)).^2);
[~,iD] = sort(D);
x1_n = X(iD(1:N));
y1_n = Y(iD(1:N));
d1_n = round(D(iD(1:N)),3);
z1_n = [z1-0.1:-0.1:z1-0.1*N]'; %contenuto in C.D. [g/Kg] nei punti contigui
hold on
scatter(x(1),y(1),450,'.','r')
scatter(X,Y,16,[.45 .45 .45],'filled')
scatter(x1_n,y1_n,25,'green','filled')
hold off
axis([0,7,0,7])
box on
xlabel('Easting [m]','FontWeight','b','FontSize',12)
ylabel('Northing [m]','FontWeight','b','FontSize',12)
title({'Fissiamo il punto P_1 e calcoliamo distanza & varianza'; ...
      ['[N=',num2str(N),' punti selezionati]']},'FontWeight','b','FontSize',14)
```

Fissiamo il punto P_1 e calcoliamo distanza & varianza



Zoom in P_1

```
In [26]: figure
hold on
% plot of N neighbour locations to P1:
scatter(x1_n,y1_n,41,'green','filled'),axis([0,7,0,7])
xlabel('Easting [m]'),ylabel('Northing [m]')
for ii = 1:N
    %pi = ['P_',num2str(ii), '(' ,num2str(x1_n(ii)), ', ',num2str(y1_n(ii)), ') '];
    pi = ['p_',num2str(ii)];
    plot([x(1),x1_n(ii)], [y(1),y1_n(ii)], 'g:');
    text(abs(x(1)+x1_n(ii))/2,abs(y(1)+y1_n(ii))/2,...
        ['d=',num2str(d1_n(ii))],...
        'color','g','background','black','VerticalAlignment',...
        'bottom','FontWeight','b','FontSize',10)
    text(x1_n(ii),y1_n(ii),[num2str(z1_n(ii)), ' [g/Kg]'], ...
        'HorizontalAlignment','center','VerticalAlignment','top',...
        'FontWeight','b','FontSize',10)
    text(x1_n(ii),y1_n(ii),pi,'HorizontalAlignment','left',...
        'VerticalAlignment','bottom','FontWeight','b','color','red')
end
% plot P1:
scatter(x(1),y(1),450,'.','r'),axis([0,7,0,7])
xlabel('Easting [m]'),ylabel('Northing [m]')
text(x(1),y(1),['C_{org} = ',num2str(z1),' [g/Kg]'],...
    'HorizontalAlignment','left',...
    'VerticalAlignment','bottom','FontWeight','b','color','red')
text(x(1),y(1),P1,'HorizontalAlignment','right','VerticalAlignment',...
    'top','FontWeight','b','color','red')
hold off
axis([min(x1_n)/1.3,max(x1_n)*1.2,min(y1_n)/1.3,max(y1_n)*1.2])
box on
xlabel('Easting [m]','FontWeight','b','FontSize',12)
ylabel('Northing [m]','FontWeight','b','FontSize',12)
title('Distanze dei punti contigui da P_1',...
    'FontWeight','b','FontSize',14)
```



Calcoliamo le distanze geospaziali vs varianze per le N coppie di punti rappresentate nel grafico sopra: $h_{1i} = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}$ $\gamma(h_{1i}) = (z(x_1 + h_{1i}) - z(x_1))^2$

```
In [15]: gamma = (z1_n - z1).^2;
          fprintf('%8s, %8s\n', 'Distanza', 'Varianza')
          for ii=1:numel(gamma) fprintf('%8.3f, %8.3f\n', d1_n(ii), gamma(ii)); end
```

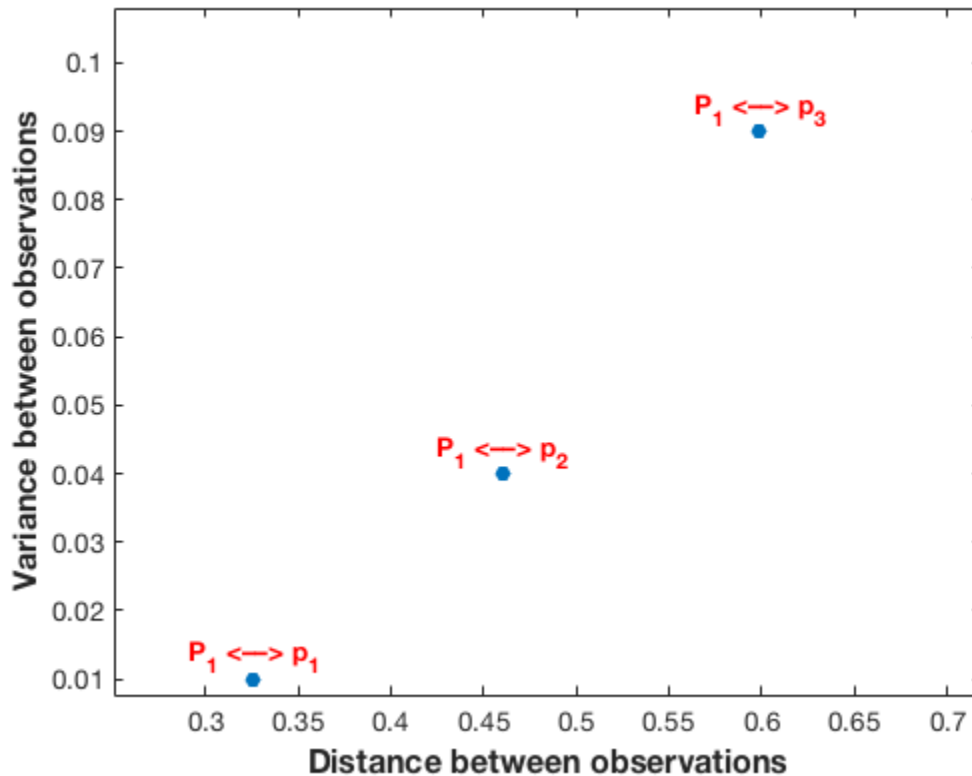
```
Distanza, Varianza
0.326,    0.010
0.460,    0.040
0.599,    0.090
```

```
In [16]: scatter(d1_n, gamma, 'filled')
          xlabel('Distance between observations', 'FontWeight', 'b', 'FontSize', 12)
          ylabel('Variance between observations', 'FontWeight', 'b', 'FontSize', 12)
          %axis([0,1,0,0.125])
          box on
          for ii = 1:N
              text(d1_n(ii), gamma(ii), ['P_1 <> p_', num2str(ii)], ...
                  'HorizontalAlignment', 'center', ...
```

```

        'VerticalAlignment','bottom','FontWeight','b','color','red')
end
axis([min(d1_n)/1.3,max(d1_n)*1.2,min(gamma)/1.3,max(gamma)*1.2])

```



0.1.5 Estendiamo il calcolo a tutti i punti di campionamento

(cioè collegando ciascun punto con tutti gli altri...)

```

In [17]: % DISTANCES
[X1,X2] = meshgrid(X);
[Y1,Y2] = meshgrid(Y);
D = sqrt((X1 - X2).^2 + (Y1 - Y2).^2);

% SEMIVARIANCE
[Z1,Z2] = meshgrid(Z);
Gamma = (Z1 - Z2).^2;

% LOWER-LEFT TRIANGLE
indx = 1 : length(Z);
[C,R] = meshgrid(indx);
I = R > C;

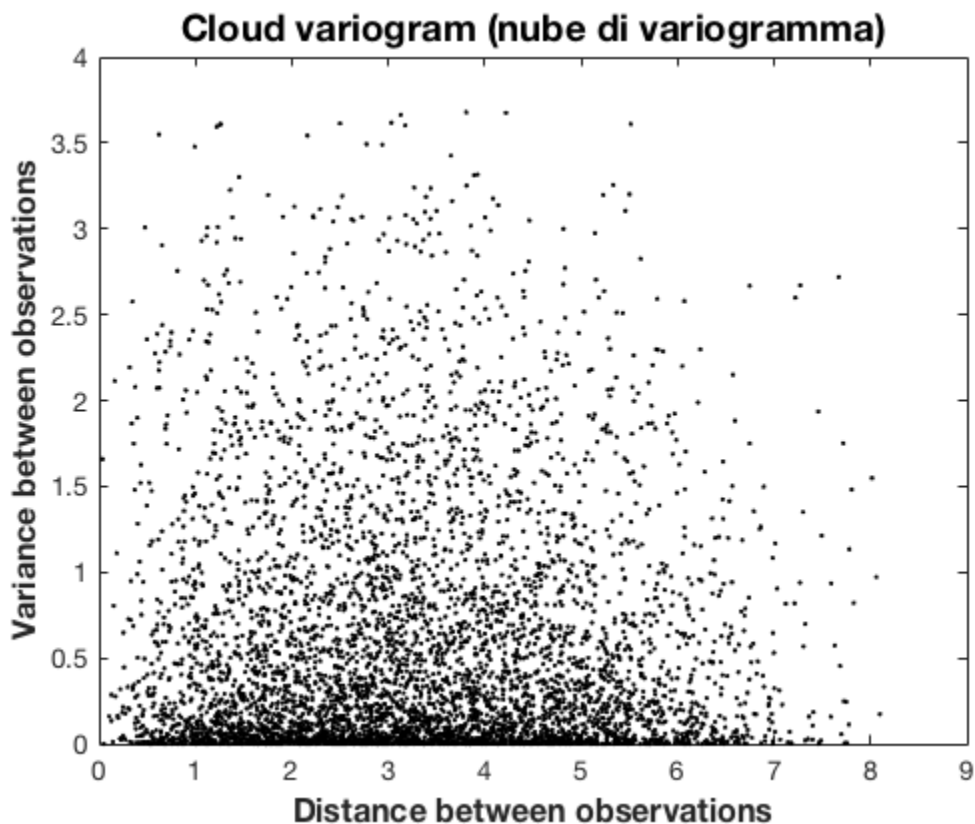
```

```
% we only take I, because values are duplicated over the two triangles:
r=23;c=2;
fprintf( 'D(%d,%d) = %7.4f\nD(%d,%d) = %7.4f\n', r,c,D(r,c), c,r,D(c,r) )
```

```
D(23,2) = 5.5287
```

```
D(2,23) = 5.5287
```

```
In [18]: plot(D(I),Gamma(I),'Marker','.','LineStyle','none',...
           'MarkerFaceColor',[.6 .6 .6],'MarkerEdgeColor','k','MarkerSize',1)
xlabel('Distance between observations','FontWeight','b','FontSize',12)
ylabel('Variance between observations','FontWeight','b','FontSize',12)
title('Cloud variogram (nube di variogramma)',...
       'FontWeight','b','FontSize',14)
```



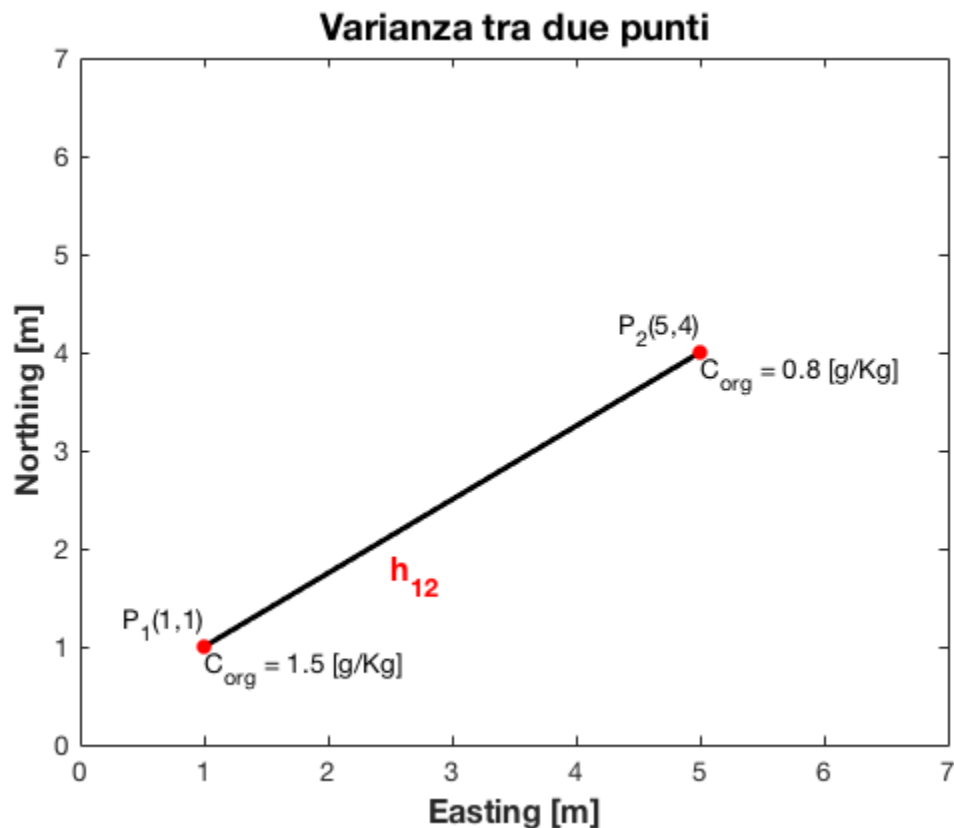
0.1.6 Costruzione del V A R I O G R A M M A sperimentale

$$\gamma(h) = \sum_{i=1}^{n(h)} \frac{(z(x_i + h) - z(x_i))^2}{n(h)}$$

Per costruire il variogramma sperimentale calcoliamo due parametri di base seguendo regole empiriche (rule of thumb).

```
In [19]: hold on
plot(x,y,'-k','LineWidth',2)
scatter(x,y,255,'.','r'),axis([0,7,0,7])
hold off
xlabel('Easting [m]','FontWeight','b','FontSize',12)
ylabel('Northing [m]','FontWeight','b','FontSize',12)
title('Varianza tra due punti','FontWeight','b','FontSize',14)

text(x(2)/2,y(2)/2,['\color{red}h_{12}'],'HorizontalAlignment','left',...
      'VerticalAlignment','top','FontWeight','b','FontSize',12)
text(x(1),y(1),P1,'HorizontalAlignment','right',...
      'VerticalAlignment','bottom')
text(x(2),y(2),P2,'HorizontalAlignment','right',...
      'VerticalAlignment','bottom')
text(x(1),y(1),['C_{org} = ',num2str(z1),' [g/Kg]'],...
      'HorizontalAlignment','left','VerticalAlignment','top')
text(x(2),y(2),['C_{org} = ',num2str(z2),' [g/Kg]'],...
      'HorizontalAlignment','left','VerticalAlignment','top')
box on
```



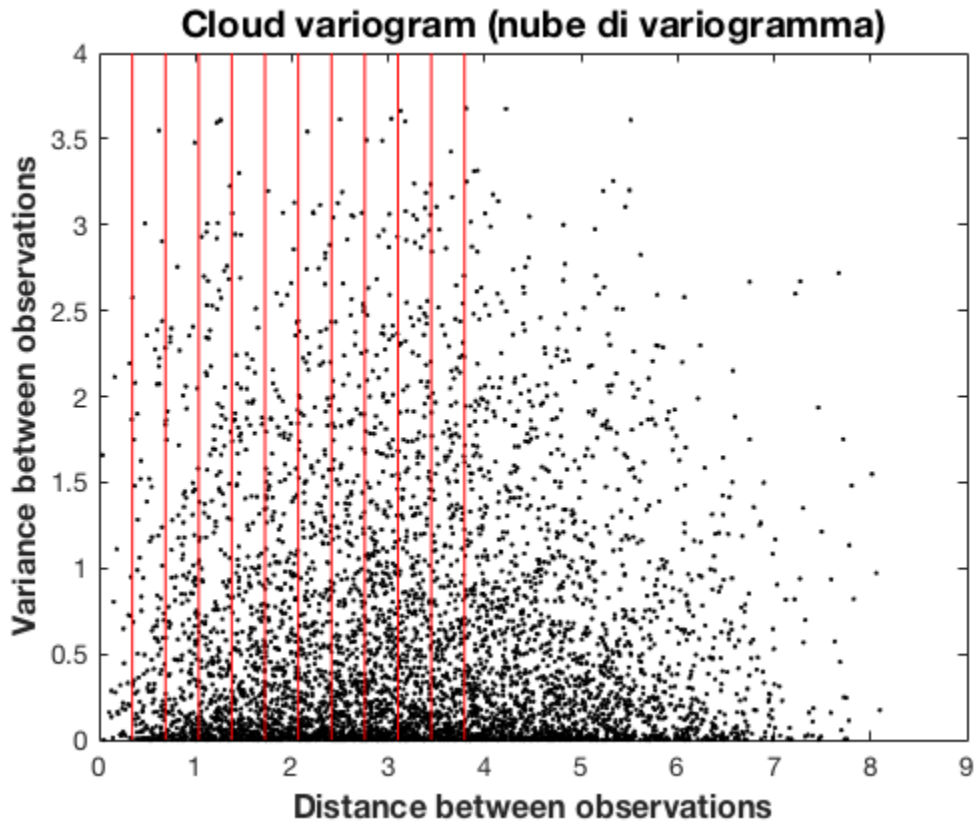
massima distanza (h_{max}) tra i 2 punti della coppia separata da $h \rightarrow \max(\frac{D}{2})$
dimensione del lag (*DEFINIRE IL LAG!) $\rightarrow \text{mean}(\min(D))$

```
In [20]: hmax = max(D(:))/2; % rule of thumb #1
D2 = D.*(diag(X*NaN)+1);
lag = mean(min(D2)); % rule of thumb #2
max_lags = floor(hmax/lag);

LAGS = ceil(D/lag);
for i = 1 : max_lags
    SEL = (LAGS == i);
    PN(i) = sum(sum(SEL == 1))/2; % numero di coppie di punti in quel LAG
    DE(i) = mean(mean(D(SEL))); % distanza media a quel LAG
    GE(i) = mean(mean(Gamma(SEL))); % varianza media a quel LAG
end
```

Vediamo graficamente la nube di variogramma sperimentale con la separazione in lag:

```
In [21]: hold on
plot(D(I),Gamma(I),'Marker','.', 'LineStyle','none',...
     'MarkerFaceColor',[.6 .6 .6], 'MarkerEdgeColor','k', 'MarkerSize',1)
xlabel('Distance between observations', 'FontWeight','b', 'FontSize',12)
ylabel('Variance between observations', 'FontWeight','b', 'FontSize',12)
title('Cloud variogram (nube di variogramma)',...
     'FontWeight','b', 'FontSize',14)
for ii = 1:max_lags
    clag = lag*ii;
    plot([clag,clag],[0,4], '-r') % max(Gamma(:))
end
hold off, box on
```

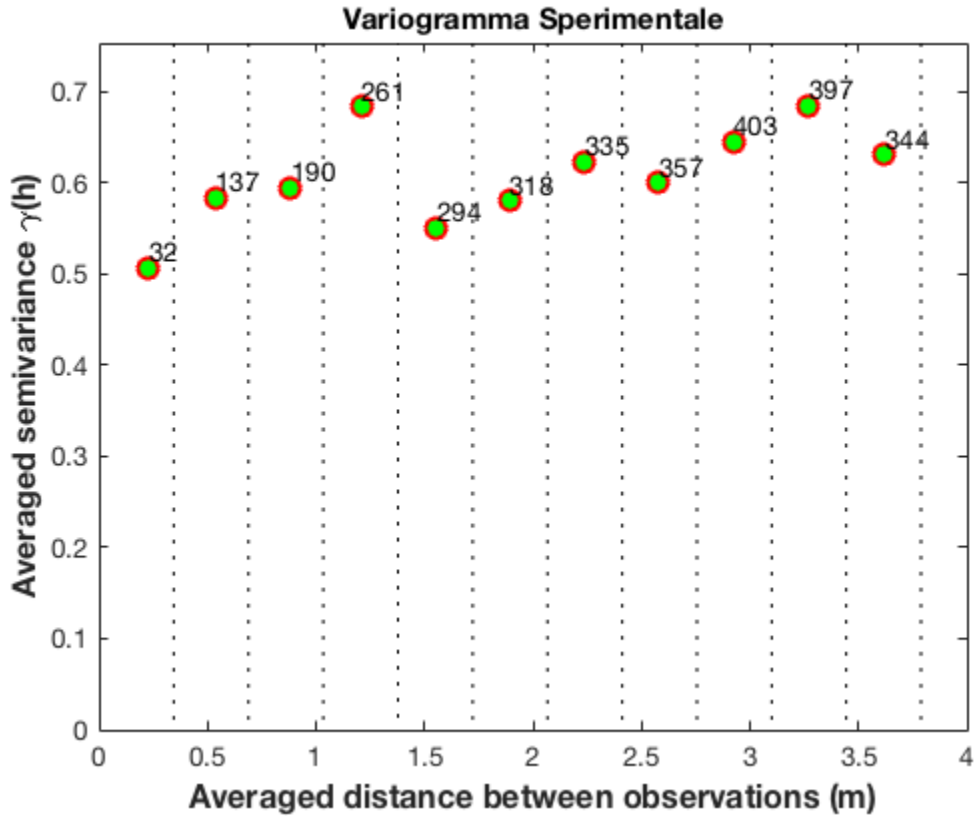
Vediamo graficamente il variogramma sperimentale:

```
In [22]: plot(DE,GE,'Marker','o','LineStyle','none',...
           'MarkerFaceColor','green','MarkerEdgeColor','r',...
           'MarkerSize',8)
var_z = var(Z);
b = [0 max(DE)];
c = [var_z var_z];
hold on
%plot(b,c,'-k')
yl = 1.1 * max(GE);
ylim([0 yl])
xlabel('Averaged distance between observations (m)',...
       'FontWeight','b','FontSize',12)
ylabel('Averaged semivariance \gamma(h)',...
       'FontWeight','b','FontSize',12)
for ii = 1:max_lags
    clag = lag*ii;
    plot([clag,clag],[0,max(Gamma(:))],':k')
end
hold off
```

```

for ii=1:numel(PN)
    text(DE(ii),GE(ii),num2str(PN(ii)),'Verticalalignment','bottom',...
        'HorizontalAlignment','left')
end
title('Variogramma Sperimentale')

```



Arriviamo alla costruzione della formula del variogramma sperimentale: (devo dividere in blocchi e spiegarli uno per uno) $\gamma(h) = \sum_{i=1}^{n(h)} \frac{(z(x_i + h) - z(x_i))^2}{n(h)}$

```

In [23]: fprintf( '%8s, %8s, %8s, %8s\n', 'LAG', 'DIST', 'GAMMA', 'Ncoppie' )
        for ii=1:max_lags
            fprintf( '%8.2f, %8.2f, %8.2f, %8d\n', lag*ii, DE(ii), GE(ii), PN(ii) )
        end

```

LAG,	DIST,	GAMMA,	Ncoppie
0.34,	0.23,	0.51,	32
0.69,	0.54,	0.58,	137
1.03,	0.88,	0.59,	190
1.38,	1.21,	0.68,	261
1.72,	1.56,	0.55,	294
2.07,	1.89,	0.58,	318

2.41,	2.24,	0.62,	335
2.76,	2.58,	0.60,	357
3.10,	2.92,	0.65,	403
3.44,	3.27,	0.68,	397
3.79,	3.62,	0.63,	344

0.2 Analizziamo i principali caratteri di un variogramma

```
In [2]: %nugget      %sill      %range
        c0=0.5;      c1=4;      a0=1;
        % distance
        d = 0:0.01:3;

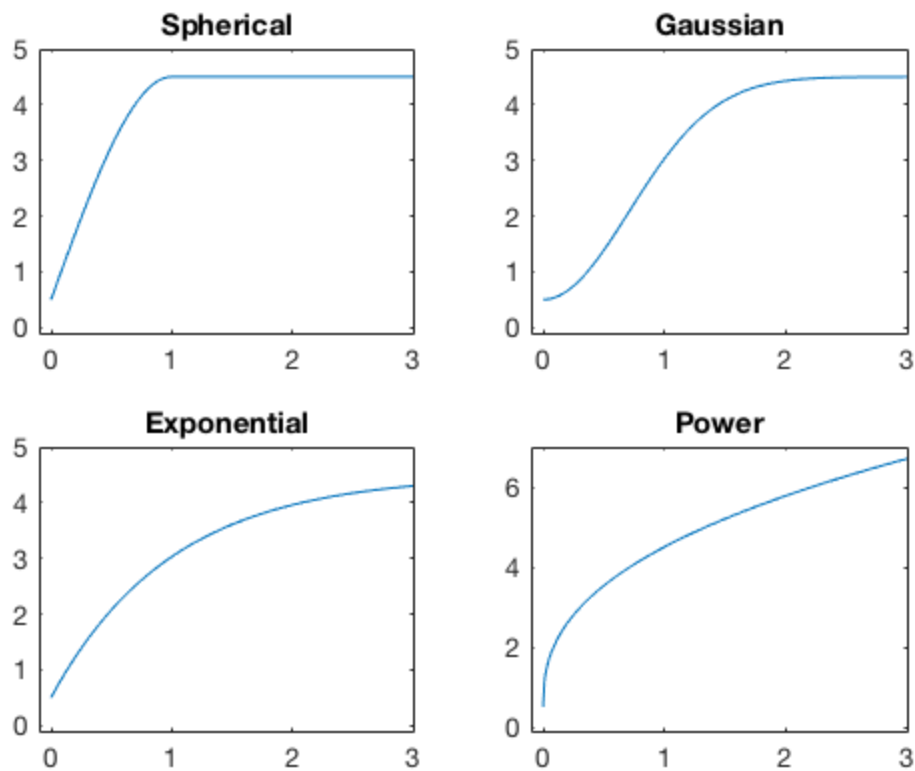
In [3]: % Spherical
        for h = 1 : numel(d)
            if d(h)<= a0
                g_sph(h) = c0 + c1*( ((3/2)*(d(h)/a0)) - ((1/2)*((d(h)/a0).^3)) );
            elseif d(h) > a0
                g_sph(h) = c0 + c1;
            end
        end
        subplot(221)
        plot(d,g_sph); xlim([-0.1,3]); ylim([-0.1, 5])
        title("Spherical")

        % Gaussian
        for h = 1 : numel(d)
            g_gau(h) = c0 + c1*( 1 - exp(-(d(h)^2)/(a0^2)) );
        end
        subplot(222)
        plot(d,g_gau); xlim([-0.1,3]); ylim([-0.1, 5])
        title("Gaussian")

        % Exponential
        for h = 1 : numel(d)
            g_exp(h) = c0 + c1*( 1 - exp(-(d(h)/a0)) );
        end
        subplot(223)
        plot(d,g_exp); xlim([-0.1,3]); ylim([-0.1, 5])
        title("Exponential")

        % Power
        a0_p = 0.4;
        for h = 1 : numel(d)
            g_pow(h) = c0 + c1*( (d(h)^a0_p) );
        end
        subplot(224)
```

```
plot(d,g_pow); xlim([-0.1,3]); ylim([-0.1, 7])
title("Power")
```



```
In [4]: % Spherical
for h = 1 : numel(d)
    if d(h) <= a0
        g_sph(h) = c0 + c1*( ((3/2)*(d(h)/a0)) - ((1/2)*((d(h)/a0).^3)) );
    elseif d(h) > a0
        g_sph(h) = c0 + c1;
    end
end
c = c0 + c1;
c_b = c - 0.10;% c bottom for plot
c_t = c + 0.15;% c top for plot
col_type = '-k';
hold on
plot(d,g_sph,'-g','LineWidth',4)
plot([a0-.02,a0+.02],[c-.05,c+.05],'-r','LineWidth',2)% X
plot([a0-.02,a0+.02],[c+.05,c-.05],'-r','LineWidth',2)% X
% nugget
plot([0-.02,0] , [c0-.05,c0] , col_type)
```

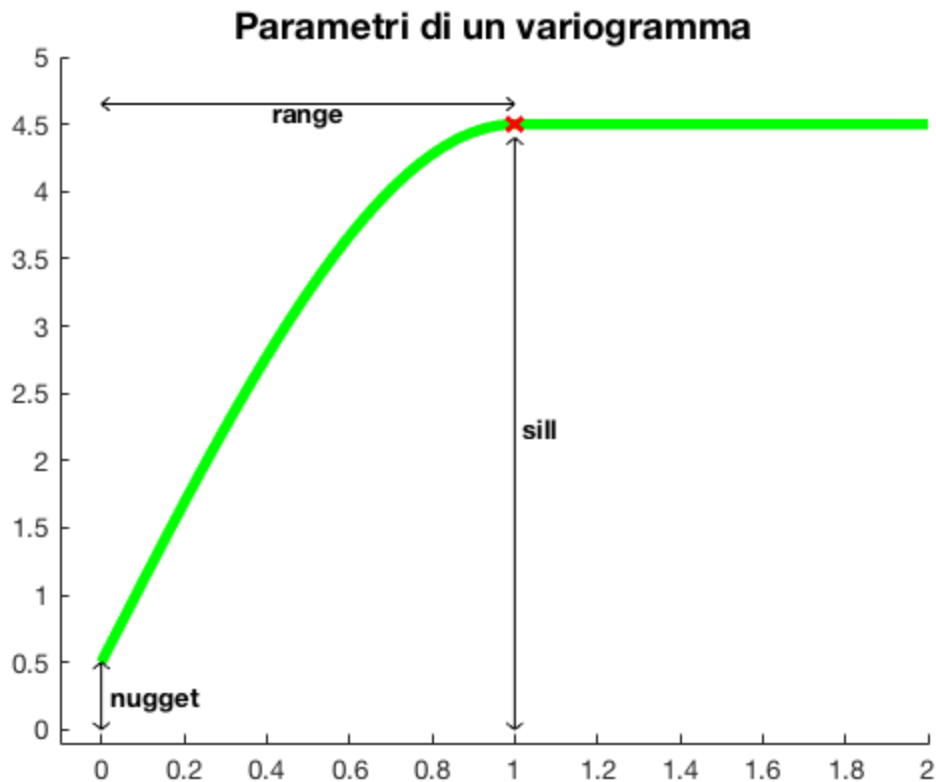
```

plot([0,0+.02]      , [c0,c0-.05]  ,col_type)
plot([0-.02,0]      , [0+.05,0]    ,col_type)
plot([0,0+.02]      , [0,0+.05]    ,col_type)
plot([0,0]          , [0,c0]       ,col_type)
% sill
plot([a0-.02,a0]    , [c_b-.05,c_b],col_type)
plot([a0,a0+.02]    , [c_b,c_b-.05],col_type)
plot([a0-.02,a0]    , [0+.05,0]    ,col_type)
plot([a0,a0+.02]    , [0,0+.05]    ,col_type)
plot([a0,a0]        , [0,c_b ]     ,col_type)
% range
plot([0,0+.02]      , [c_t,c_t+.05],col_type)
plot([0,0+.02]      , [c_t,c_t-.05],col_type)
plot([a0-.02,a0]    , [c_t+.05,c_t],col_type)
plot([a0-.02,a0]    , [c_t-.05,c_t],col_type)
plot([0,a0]         , [c_t,c_t ]   ,col_type)
hold off

xlim([-0.1,2.0]); ylim([-0.1, 5])

text(0+.02,c0/2,"nugget","FontWeight",'b')
text(a0+.02,c/2,"sill","FontWeight",'b')
text(a0/2,c_t -.05 ,"range","FontWeight",'b','HorizontalAlignment','center')
title("Parametri di un variogramma","FontWeight",'b','FontSize',14)

```



0.3 Facciamo alcuni esempi

Vedere documento successivo.