Spatial autocorrelation 3: Kriging with covariates

Michael Noonan

DATA 589: Spatial Statistics

Table of contents

Review

Review

Last lecture we learned how to fit semi-variograms to spatial data, and leverage the information contained in autocorrelation structures to learn about processes.

We also saw how Kriging can use the information contained in these models to interpolate spatially referenced data, but that it is sensitive to model specification, spatially constrained, and slow.

Ordinary Kriging leverages information contained in the autocorrelation structure to make predictions, but doesn't use any information from covariates.

Today we will cover a suite of tools for making predictions with covariates (i.e., co-Kriging, regression Kriging, and regression with autocorrelated errors).

Co-Kriging

Motivation

When Kriging, we are trying to predict the values of some target variable, but we could have just measured the variable everywhere we wanted to predict.

In reality, this is rarely possible because its costly and time consuming to collect the data, which means we have typically have few observations.

... but if there is another variable that is cheaper/easier to measure, and covaries with our target variable, then we can collect more observations and leverage their information to improve our estimates.

Co-Kriging

Co-Kriging is an extension of ordinary Kriging in which additional observed variables are used to improve the interpolation of the variable of interest.

Co-Kriging does **not** require that the secondary information is available at all prediction locations.

The co-variable may be measured at the same points as the target (co-located samples), at other points, or both.

Cross-variogram

Where Kriging relied on the variogram to make predictions, co-kriging relies on the cross-variogram.

$$\hat{\gamma}_{AB}(h) = \frac{1}{2N(h)} \sum_i^n \sum_j^m \{Z_A(x_i) - Z_A(x_j)\}\{Z_B(x_i) - Z_B(x_j)\}$$

If differences between point-pairs of variable A are associated with differences between point-pairs of variable B, they will have a strong cross-correlation.

Co-Kriging

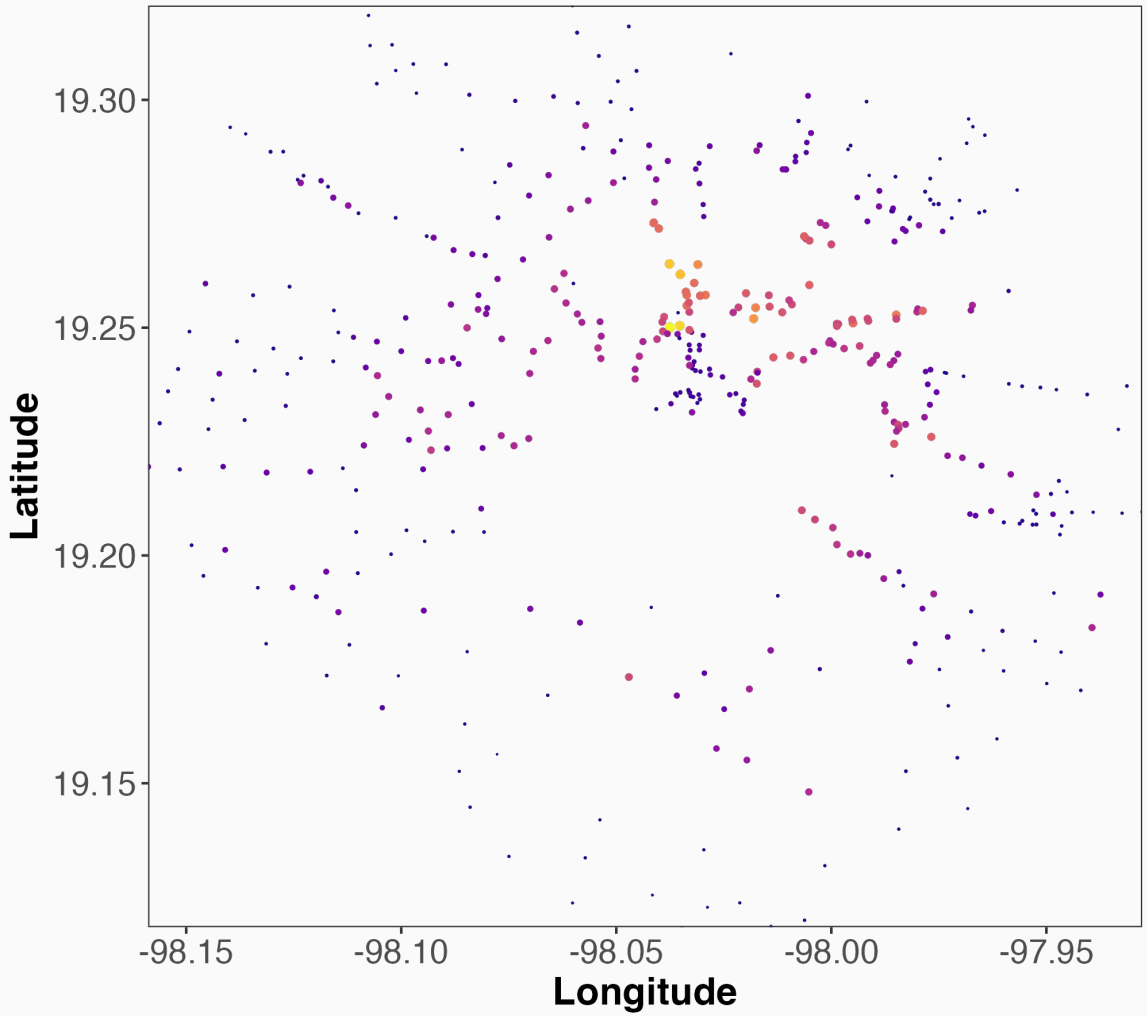
Prediction of the target variable at unknown locations s_0 is computed as a linear combination of n locations of the target variable A and p locations of a co-variable B.

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z_A(s_i) + \sum_{j=1}^{p} \alpha_j Z_B(s_j)$$

where λ and α are the weights for target variable and co-variable, and $\sum \lambda = 1$ and $\sum \alpha = 0$.

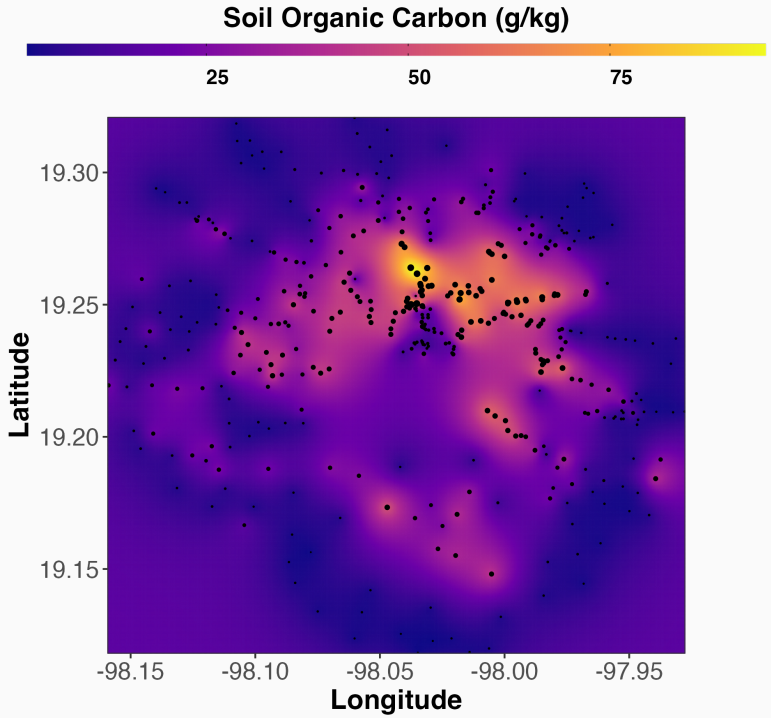
Note: the direct and cross-variograms must be modeled together, to ensure that the weights can be calculated (for more details see: Knotters *et al.*, 1995).

Today we'll work with data on soil organic carbon (SOC; in g/kg) in central Mexico from (Fusaro *et al.*, 2019)



Co-Kriging in R cont.

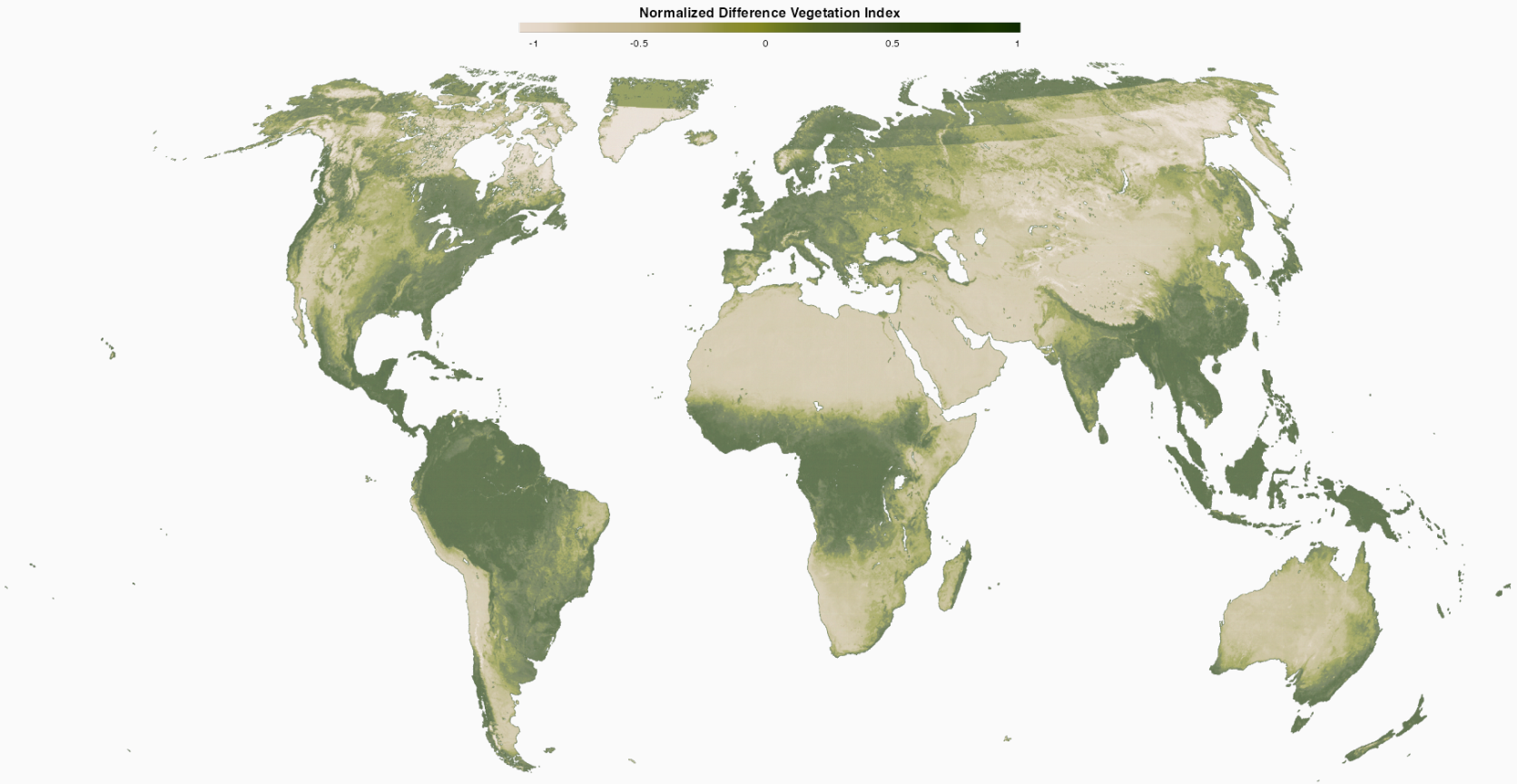
We could map the distribution of SOC via ordinary Kriging



... but what if we know SOC is related to environmental productivity? ...and we had lots of data on env. prod.

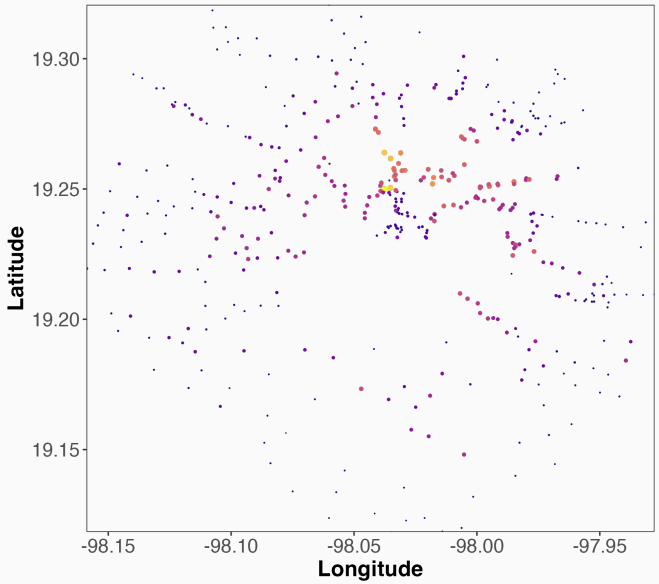
NDVI

NDVI is a satellite derived measure of environmental productivity (global, updated every 16 days).

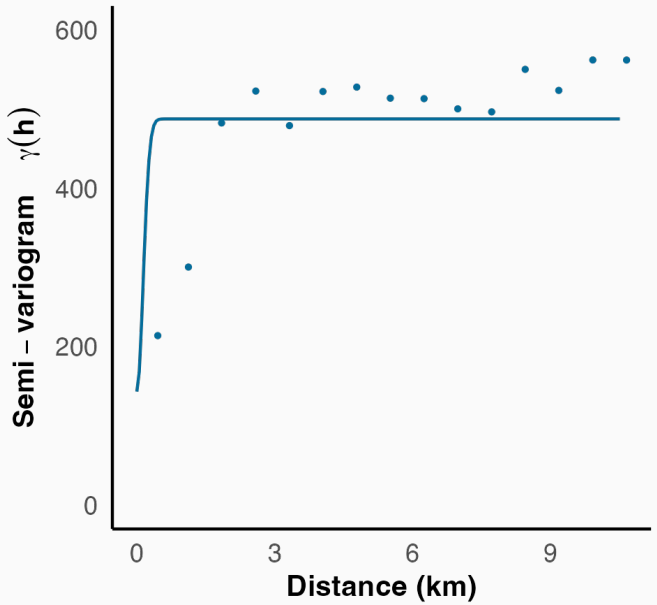


Co-Kriging in R cont.


```
# Import the SOC dataset
data <- read.csv("Datasets/Mexican_SOC.csv")

# prepare coordinates, data, and proj4string
coords <- data[, c("Long", "Lat")]
data   <- data[, 3:6]
crs    <- CRS("+proj=longlat +datum=WGS84")

# make the SpatialPointsDataFrame object
data <- SpatialPointsDataFrame(coords = coords,
                               data = data,
                               proj4string = crs
                                    )
```

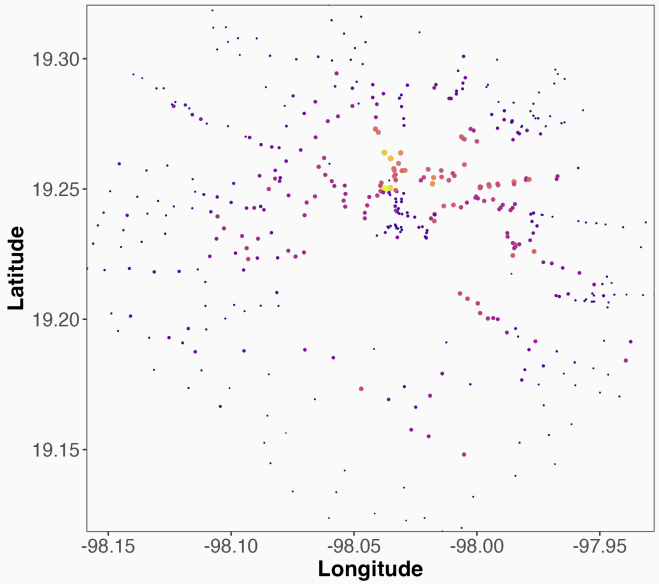


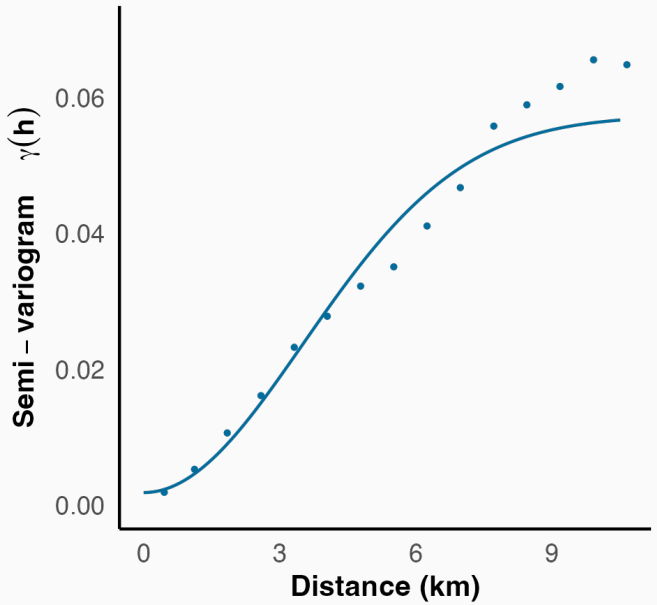

```
# Empirical variogram
soc.vg <- gstat::variogram(SOC ~ 1, data = data)

#Fit Gaussian correlation models
soc.fit <- fit.variogram(soc.vg, vgm("Gau"))

soc.fit
  model    psill     range
1   Nug 143.3185 0.0000000
2   Gau 344.1283 0.1913027
```



Co-Kriging in R cont.

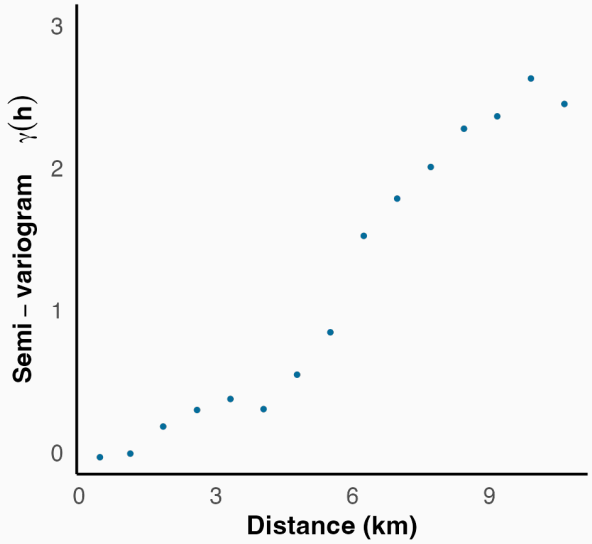

```
# Empirical variogram for NDVI
ndvi.vg <- variogram(NDVI ~ 1, data = data)


#Fit Gaussian correlation models
ndvi.fit <- fit.variogram(ndvi.vg, vgm("Gau"))


ndvi.fit

  model       psill    range
1   Nug 0.001858227  0.00000
2   Gau 0.055467054  4.95741
```



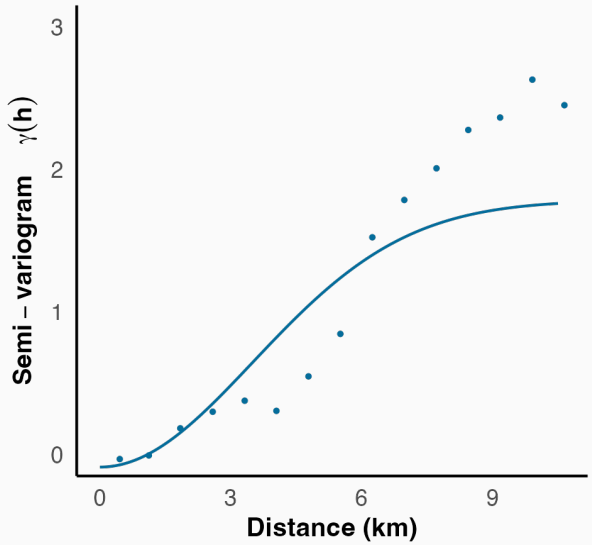


Co-Kriging in R cont.


```
#Combine into gstat object
g <- gstat(NULL, id = "SOC", form = SOC ~ 1,
     data=data)

g <- gstat(g, id = "NDVI", form = NDVI ~ 1, data
     =data)

#Estimate cross-variogram
vg.cross <- gstat::variogram(g)
```

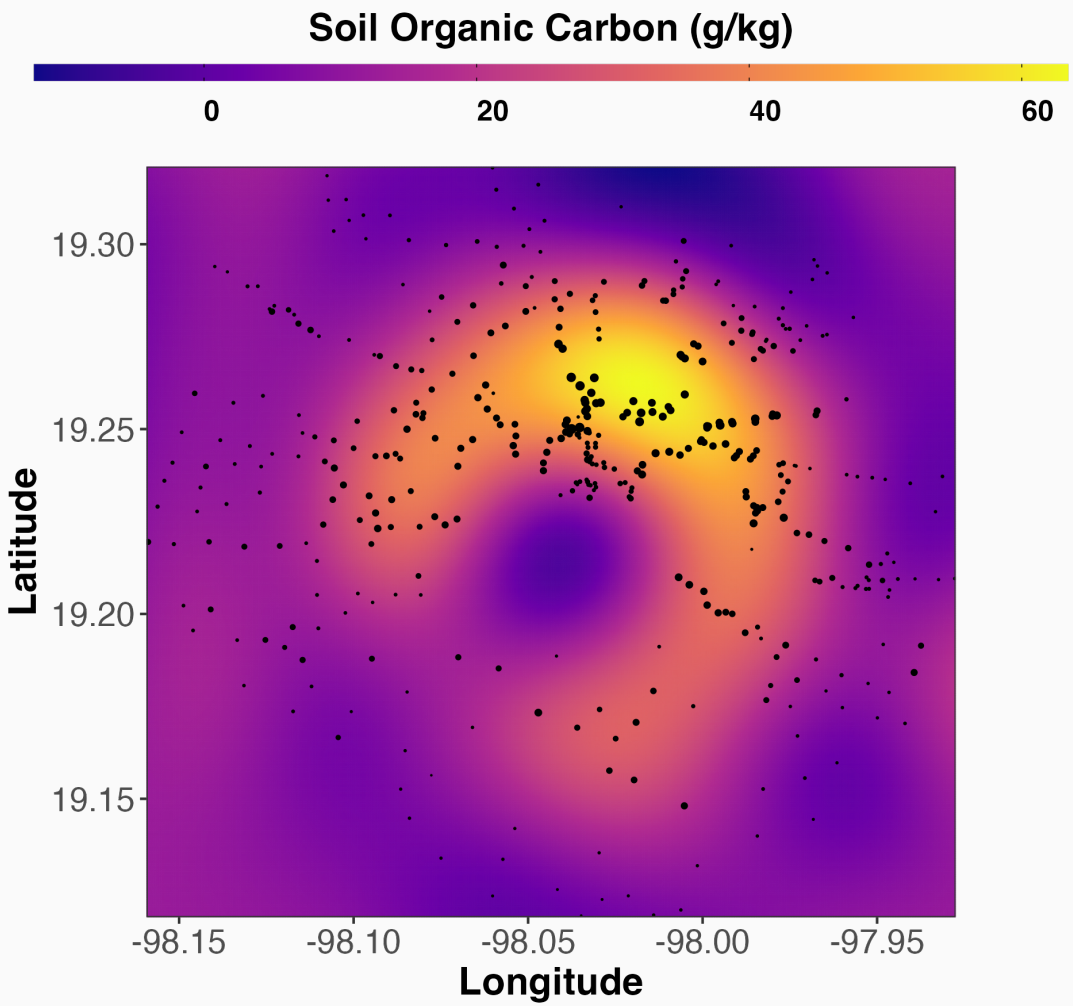



```
# Fit the model
g <- gstat(g, id = "NDVI", model = ndvi.fit,
           fill.all=T)
g <- fit.lmc(vg.cross, g)

variograms:
            model         psill    range
SOC[1]        Nug  2.603204e+02  0.00000
SOC[2]        Gau  4.074533e+02  4.95741
NDVI[1]       Nug  1.858219e-03  0.00000
NDVI[2]       Gau  5.546710e-02  4.95741
SOC.NDVI[1]   Nug  0.000000e+00  0.00000
SOC.NDVI[2]   Gau  1.869458e+00  4.95741
```



Co-Kriged SOC map


```
SOC.cokriged <- predict(g, grid)
```



Co-Kriging considerations

The direct and cross-variograms must be modeled together, so this places an additional constraint of all variables requiring the same model and range.

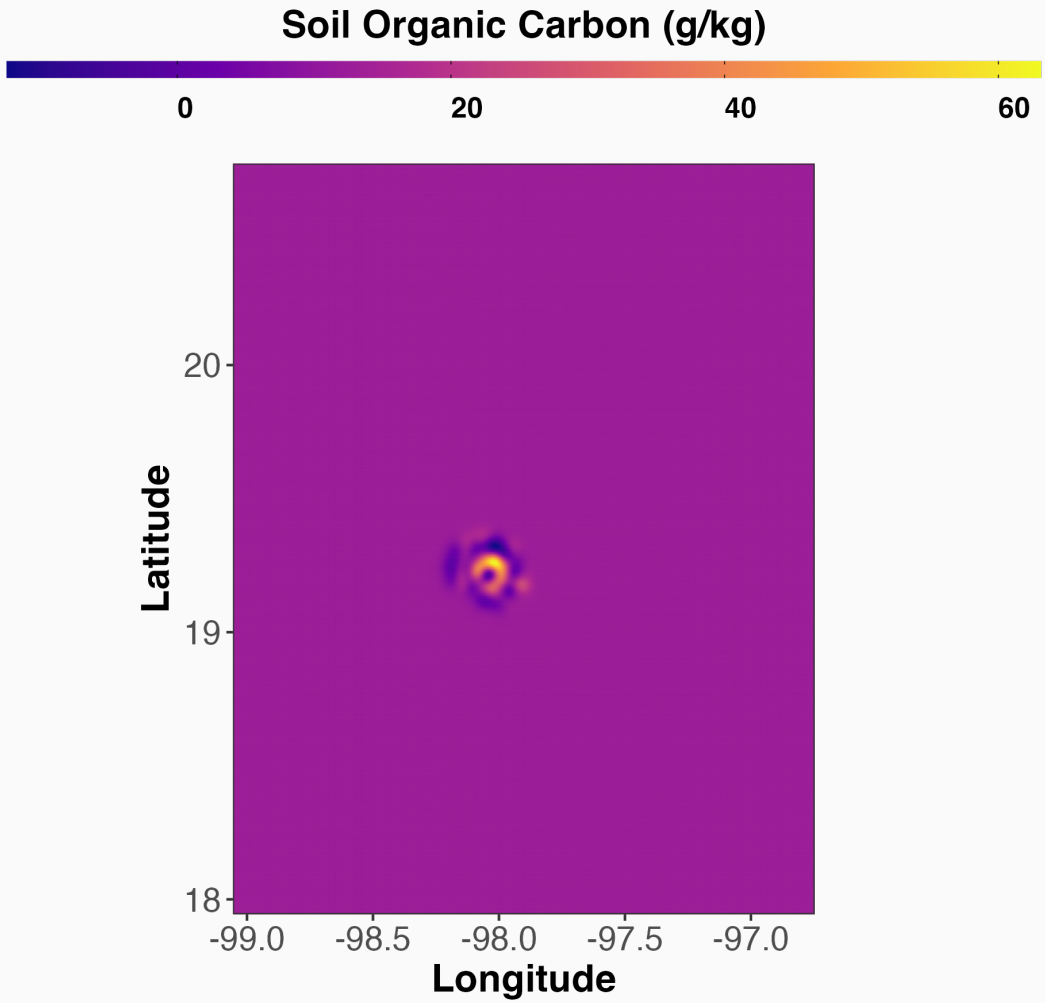
Quality of the predictions will depend on the strength of the correlations. If there is no, or only a weak correlation between the variables, co-Kriging might not be of benefit.

Even more sensitive to model misspecification.

Slower than ordinary Kriging.

Co-Kriging considerations

Still an interpolation method.

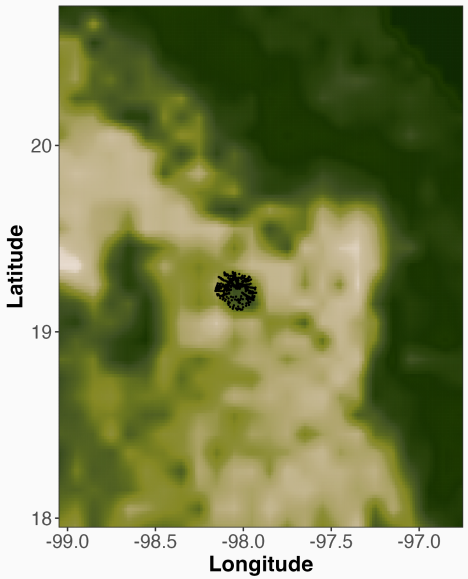


Regression Kriging

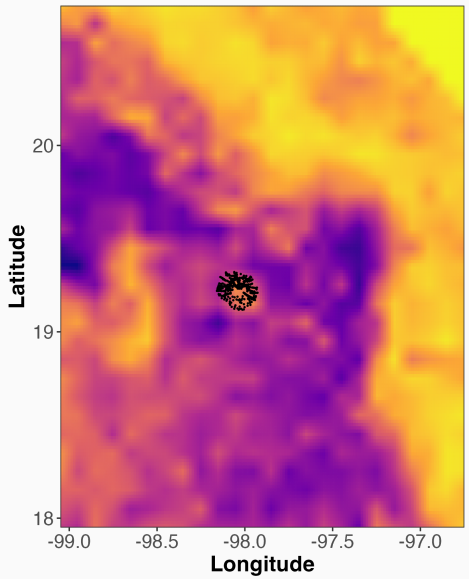
Motivation

Co-kriging relies on correlations to improve the accuracy of predictions, but if we know there's a correlation between our variables, couldn't we just use regression to make our predictions?

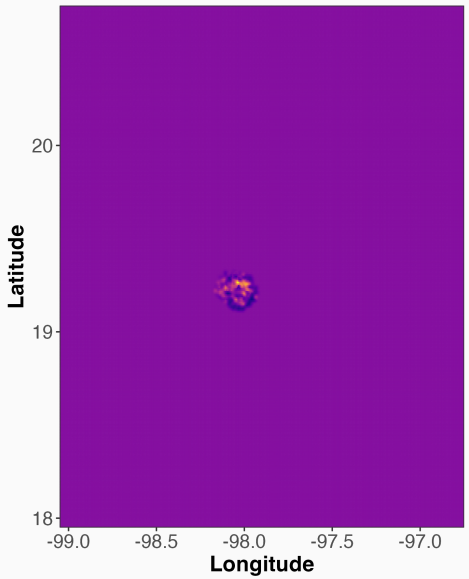
Model SOC $= \beta \times$ NDVI



Predict SOC from model



Unmodelled autocorr. in the residuals



Regression-Kriging

Regression-Kriging operates under the principle that the value of a target variable at some location s_0 can be modeled as a sum of the deterministic $m(s_0)$ and stochastic $e(s_0)$ components.

$$\hat{z}(s_0) = \hat{m}(s_0) + \hat{e}(s_0) = \sum_{k=0}^{p} \hat{\beta}_k \cdot q_k(s_0) + \sum_{i=1}^{n} \lambda_i \cdot e(s_i)$$

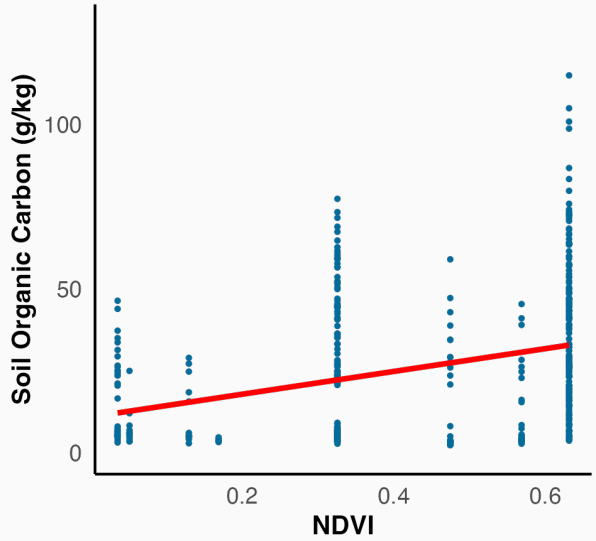
where $\hat{m}(s_0)$ is the fitted deterministic part,
$\hat{e}(s_0)$ is the Kriged residual,
$\hat{\beta}_k$ are estimated deterministic model coefficients,
λ_i are kriging weights,
and $e(s_i)$ is the residual at location s_i.

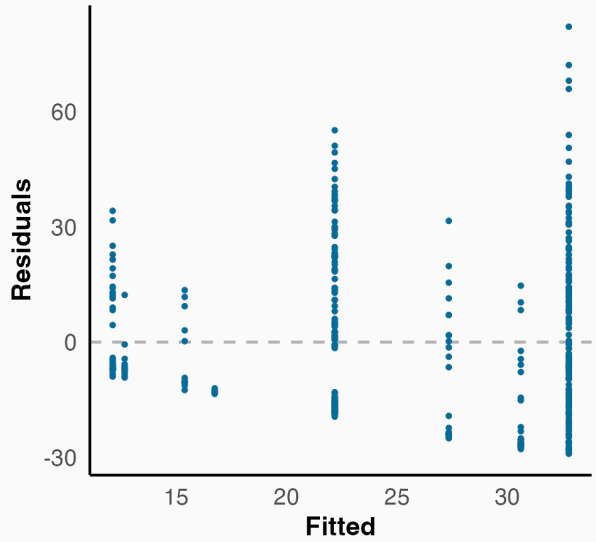
Regression-Kriging cont.

Regression-Kriging pairs a regression model's capacity to make predictions based on relationships between variables, $\hat{m}(s_0)$, with Kriging's capacity to leverage the autocorrelation structure, $\hat{e}(s_0)$.

First step is to estimate the deterministic part, $\hat{m}(s_i)$

```
# Estimate the deterministic part
m_hat <- lm(SOC ~ NDVI, data = DATA)

summary(m_hat)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   10.900      2.211   4.929 1.17e-06
      ***
NDVI          34.653      4.595   7.542 2.69e-13
      ***
---

# Store the residuals
data$residuals <- residuals(m_hat)
```



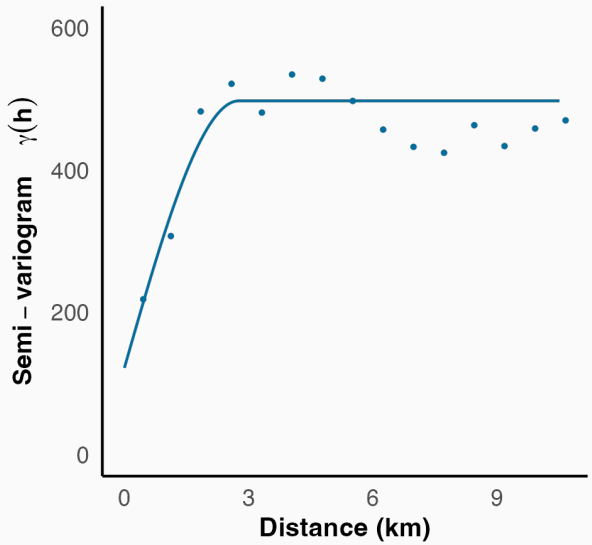


Regression-Kriging in R cont.

Next we estimate the stochastic part, $\hat{e}(s_i)$

```
# Variogram of the residuals
vg <- variogram(residuals ~ 1, data = data)


#Fit the correlation model
s_hat <- fit.variogram(vg, vgm("Sph"))


s_hat
```

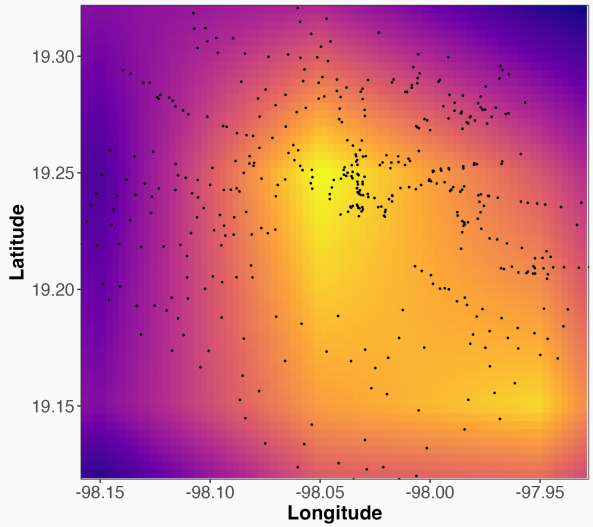
```
  model    psill    range
1   Nug 122.0617 0.000000
2   Sph 375.1581 2.762111
```



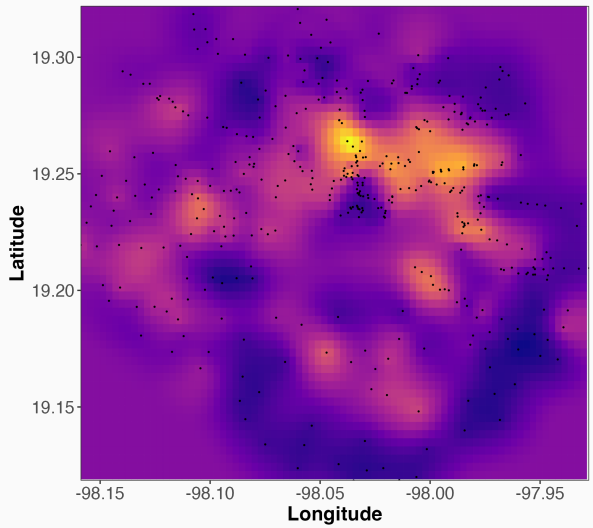
Regression-Kriging in R cont.

Then we predict from our two models.

```
#Create a dataframe of NDVI to predict from
NDVI_data <- rasterToPoints(NDVI, spatial = T)
NDVI_data <- as.data.frame(NDVI_data)
names(NDVI_data) <- c("NDVI", "Long", "Lat")

# Predict from the deterministic model
m_hat_s0 <- predict(m_hat, newdata = NDVI_data)
```




```
# Define the locations to krig over.
grid <- NDVI_data
sp::coordinates(grid) <- c("Long","Lat")
grid <- SpatialPoints(coords = grid,
                      proj4string = crs)

#Ordinary kriging of the residuals
e_hat_s0 <- krige(residuals ~ 1,
                  data,
                  newdata = grid,
                  model=residuals.fit)
```

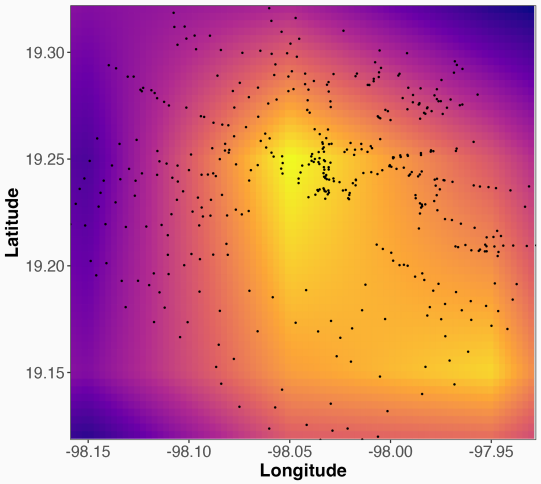


Regression-Kriging in R cont.

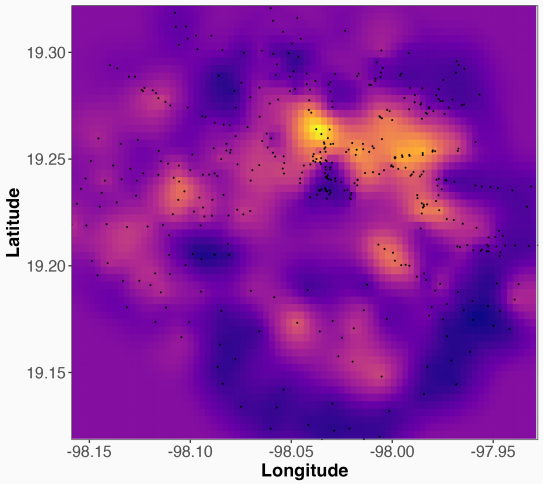
...and finish by summing the two.

```
z_hat <- m_hat_s0 + e_hat_s0@data$var1.pred
```

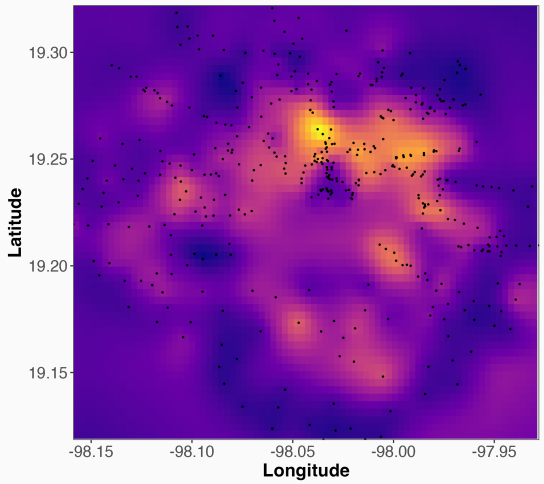
Predict from regression model



Kriged residuals



Regression-Kriged Prediction



Regression-Kriging considerations

Regression-Kriging is generally more accurate than Kriging or co-Kriging alone (for details see: Knotters *et al.*, 1995).

Regression-Kriging is a generalisation of both regression and Kriging (when there's no autocorrelation RK = regression, when there's no regression model, RK = Kriging).

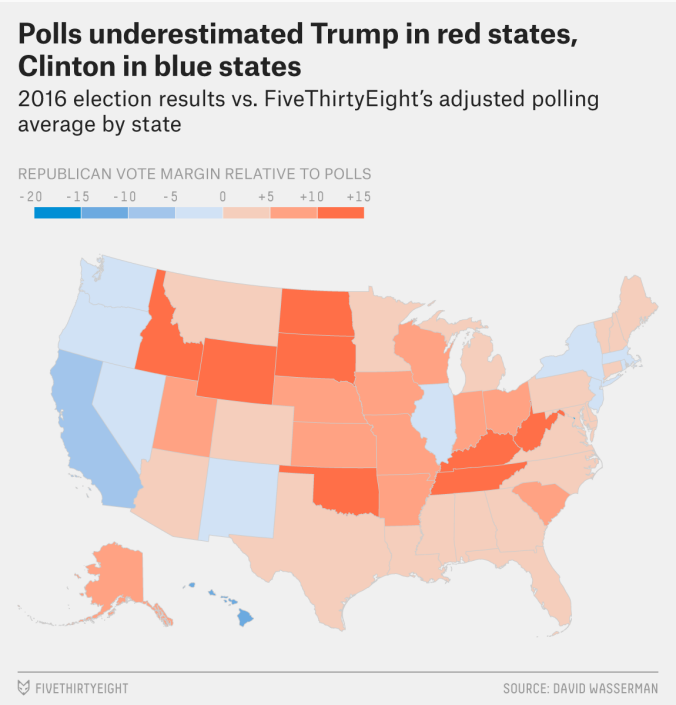
Can be used to interpolate and extrapolate (outside the range where correlations are observed the predictions are made by the regression model alone), but requires that the values of the regression model be sampled everywhere you want to predict.

Quality of the predictions depends on two models, so it is still sensitive to model misspecification.

Regression with Correlated Errors

Motivation

Everything we've covered so far has focused on predicting the value of some target variable at an unobserved location, but what if we just want to model a system using spatially collected data?



For example, when predicting election outcomes, we're not necessarily interested in predicting over space, but ignoring autocorrelation can result in poorly behaved models.

Autocorrelation and regression

The standard linear regression model has the following form:

$$y_i = \beta_0 + \beta_1 \times x_i + \varepsilon_i \qquad \varepsilon_i \sim \mathcal{N}(0,\, V) \qquad V = \sigma^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

The diagonal defines the variances. All 1s indicates homogeneity of variances.

The off-diagonals define the co-variances. The 0s indicate independence.

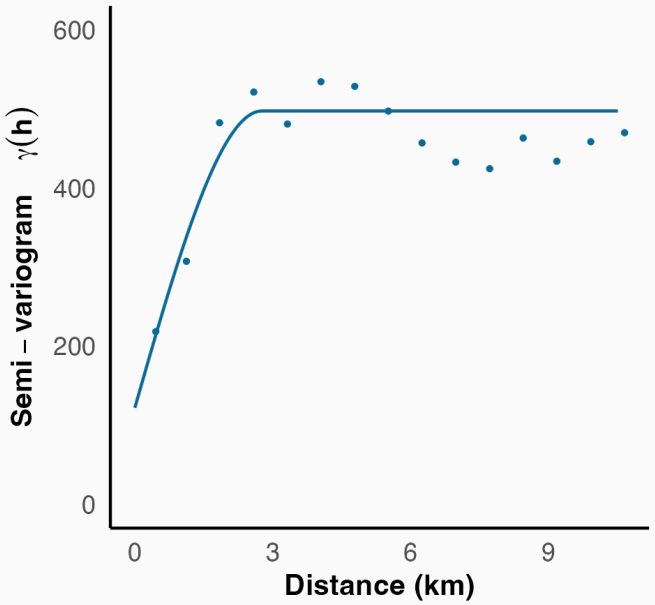
Variance-Covariance Matrix

Correcting for autocorrelation 'simply' involves identifying the autocorrelation structure of the residuals and modifying the variance-covariance matrix.

$$V = \sigma^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

When the residuals are autocorrelated, the off-diagonals $\neq 0$.

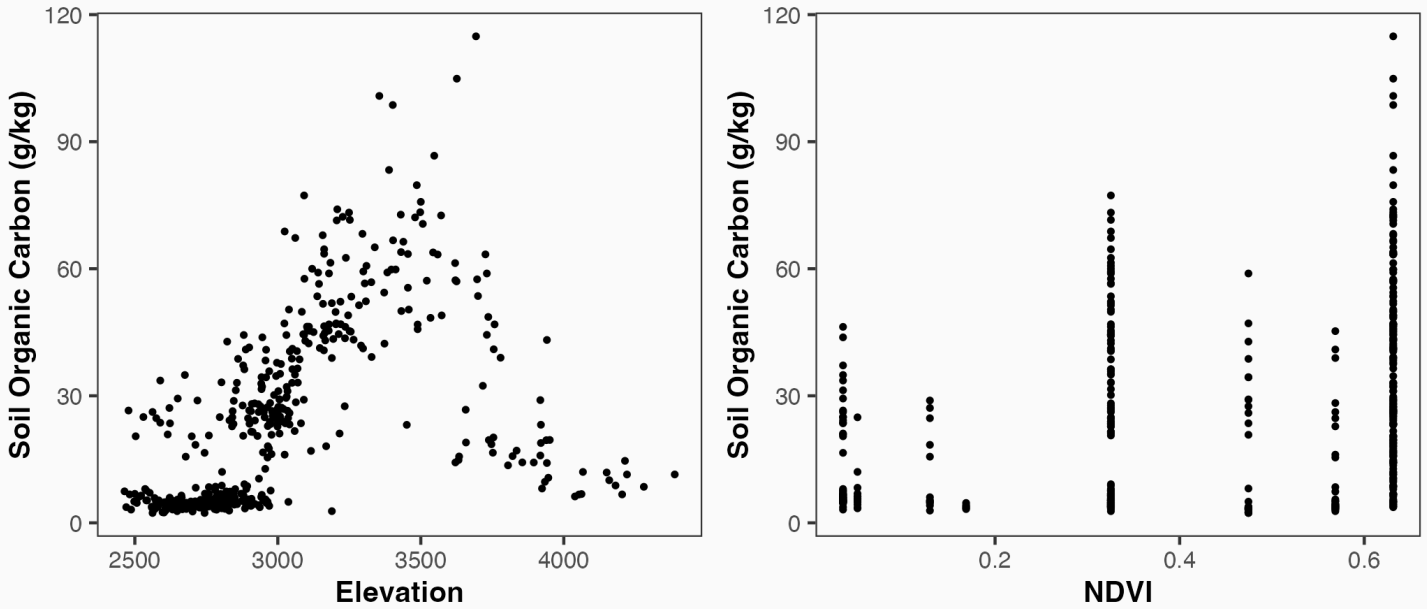
For spatial data, the correlation structure is estimated by the semi-variance model of the residuals.



Modelling SOC

Our starting point is the linear regression model:

$$\text{SOC}_i = \beta_0 + \beta_1 \text{NDVI}_i + \beta_2 \text{Elevation}_i + \beta_3 \text{Elevation}_i^2 + \varepsilon_i$$



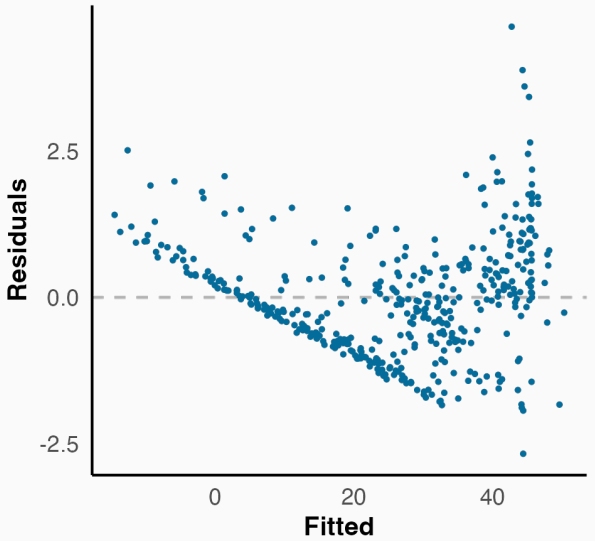
Modelling SOC in R

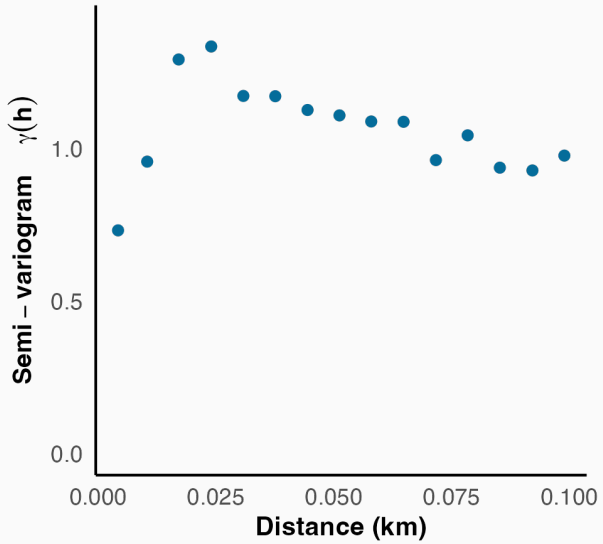
```
# Import the nlme packge for fitting the model
library(nlme)

# Fit the model using REML
FIT <- gls(SOC ~ NDVI + Altitude + I(Altitude^2), data = DATA)


# Summary of the fitted model
summary(FIT)

Generalized least squares fit by REML
  Model: SOC ~ NDVI + Altitude + I(Altitude^2)
  Data: DATA
       AIC      BIC    logLik
  3700.407 3720.795 -1845.203

Coefficients:
                  Value Std.Error    t-value p-value
(Intercept)   -750.6398  44.42102 -16.898303  0.0000
NDVI           -14.5660   4.55503  -3.197777  0.0015
Altitude         0.4626   0.02759  16.766968  0.0000
I(Altitude^2)   -0.0001   0.00000 -16.034018  0.0000
```

Autocorr. in the SOC data


```
#Spatial data frame of residuals
RES <- data.frame(res =
                    residuals(FIT,
                              type="
                                   normalized
                                   "),
                  x = DATA$Long,
                  y = DATA$Lat)

coordinates(RES) <- c("x","y")

#Calculate variogram
vg <- variogram(res ~ 1, data = RES)
```

Variogram indicates autocorrelation, so results can't be trusted.





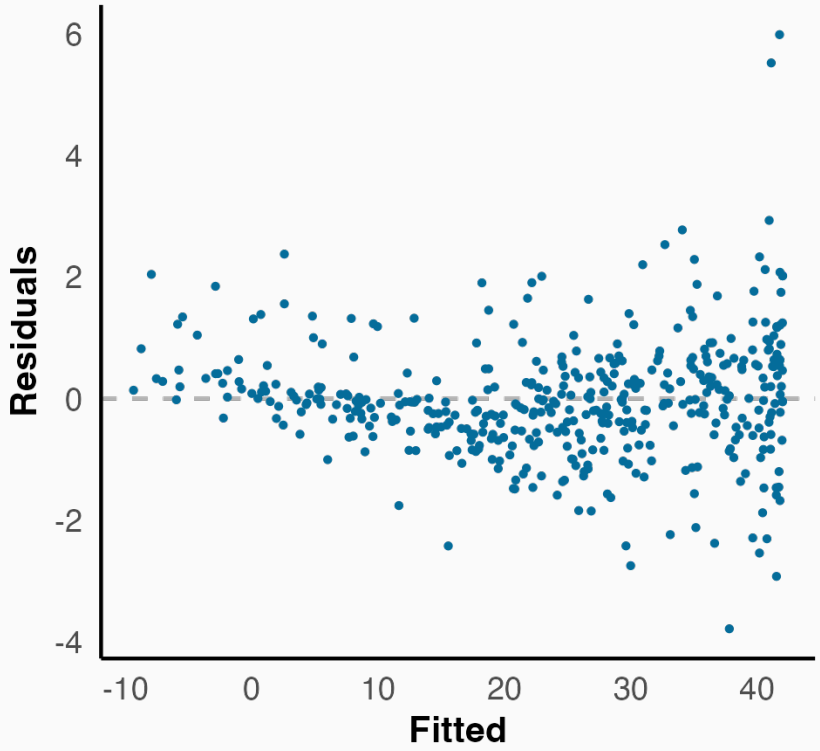
Exponential spatial correlation in R

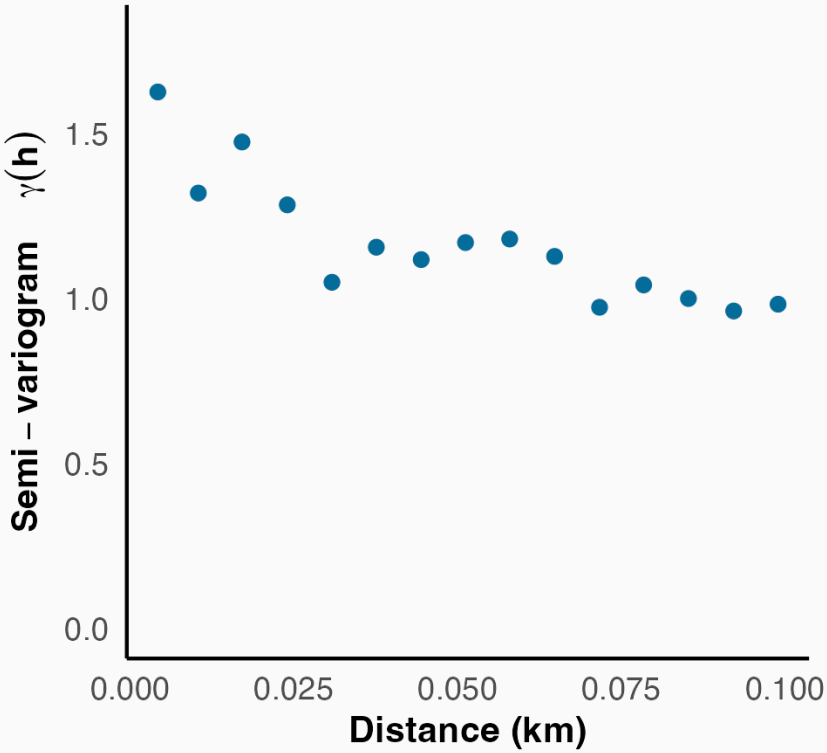
A model with exponential spatial correlation structure can be fit via the corExp() function.

```
FIT_Exp <- gls(SOC ~ NDVI + Altitude + I(Altitude^2),
               correlation = corExp(c(1, 0.001),
                                    form=~Long + Lat,
                                    nugget = TRUE),
               data = DATA)
```

```
summary(FIT_Exp)

Correlation Structure: Exponential spatial correlation
 Formula: ~Long + Lat
 Parameter estimate(s):
     range     nugget
0.01583927 0.33726949

Coefficients:
                   Value Std.Error   t-value p-value
(Intercept)   -541.3500  58.15165 -9.309281  0.0000
NDVI            -6.3930   6.23897 -1.024681  0.3061
Altitude         0.3306   0.03602  9.179753  0.0000
I(Altitude^2)    0.0000   0.00001 -8.558770  0.0000
```

Corrected model residuals





Corrected model

Original Model

```
Generalized least squares fit by REML
  Model: SOC ~ NDVI + Altitude + I(Altitude^2)
  Data: DATA
       AIC      BIC    logLik
  3700.407 3720.795 -1845.203

Coefficients:
                  Value Std.Error    t-value p-value
(Intercept)   -750.6398  44.42102 -16.898303  0.0000
NDVI           -14.5660   4.55503  -3.197777  0.0015
Altitude         0.4626   0.02759  16.766968  0.0000
I(Altitude^2)   -0.0001   0.00000 -16.034018  0.0000

 Correlation:
             (Intr) NDVI   Altitd
NDVI          0.403
Altitude     -0.998 -0.402
I(Altitude^2)  0.990  0.356 -0.996

Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.66932611 -0.70685019 -0.04261766  0.56562345  4.61986860

Residual standard error: 15.60856
Degrees of freedom: 440 total; 436 residual
```

Spatial correlation model

```
Generalized least squares fit by REML
  Model: SOC ~ NDVI + Altitude + I(Altitude^2)
  Data: DATA
       AIC      BIC    logLik
  3490.333 3518.876 -1738.166

Correlation Structure: Exponential spatial correlation
 Formula: ~Long + Lat
 Parameter estimate(s):
     range     nugget
0.01583927 0.33726949

Coefficients:
                  Value Std.Error   t-value p-value
(Intercept)   -541.3500  58.15165 -9.309281  0.0000
NDVI            -6.3930   6.23897 -1.024681  0.3061
Altitude         0.3306   0.03602  9.179753  0.0000
I(Altitude^2)    0.0000   0.00001 -8.558770  0.0000

 Correlation:
             (Intr) NDVI   Altitd
NDVI          0.153
Altitude     -0.995 -0.179
I(Altitude^2)  0.981  0.161 -0.994

Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.32734273 -0.67849720  0.03326135  0.70771455  4.89984702

Residual standard error: 15.05363
Degrees of freedom: 440 total; 436 residual
```

Take home messages

There are different options for incorporating covariates when modelling spatial data, and each has their pros and cons.

Co-Kriging allows you to incorporate covariates, and doesn't require that the samples are co-located, but places constraints on the models and only interpolates.

Regression-Kriging allows you to incorporate covariates, can extrapolate, but requires that the information on the covariates are available at the prediction locations.

Correlated error models can improve the reliability of regression models, but aren't designed for spatial predictions.

References

Fusaro, C., Sarria-Guzmán, Y., Chávez-Romero, Y.A., Luna-Guido, M., Muñoz-Arenas, L.C., Dendooven, L., Estrada-Torres, A. & Navarro-Noya, Y.E. (2019). Land use is the main driver of soil organic carbon spatial distribution in a high mountain ecosystem. *PeerJ*, 7, e7897.

Knotters, M., Brus, D. & Voshaar, J.O. (1995). A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations. *Geoderma*, 67, 227–246.