Generalized Additive Models in Spatial Statistics

Linear Models with a Twist

John Maindonald (Centre for Mathematics and Its Applications, Australian National University)

April 24, 2011

Overview

- Kriging for time series? [can this give insight?]

- Data Exploration using regression methods [c.f. Section 5.3 (pp.127–133) of *A Practical Guide to Geostatistical Mapping?*]

- GAM-based fitting of trend curves and surfaces.
  When should regression stop and kriging start?

- The uses and limits of GAM models.
  What are the uses of GAM models, in the Geostatistical armoury?

Can Kriging have Bad Effects on the Psyche?

"Surely, the endless list of kriging methods, each with an infinite set of kriged estimates, makes ore reserve and resource estimation a daunting discipline of art and science."

[Jan Merks[a], who describes kriging as voodoo science! Does Merks' cartoon perhaps hint that kriging may not be good for the psyche?]

[a] http://geostatscam.com/kriging_game.htm

Here, it will be argued that the challenge is to know when to stop trend surface fitting (regression) and start kriging (in those circumstances where kriging can improve accuracy). Why stop with a deficient trend model?

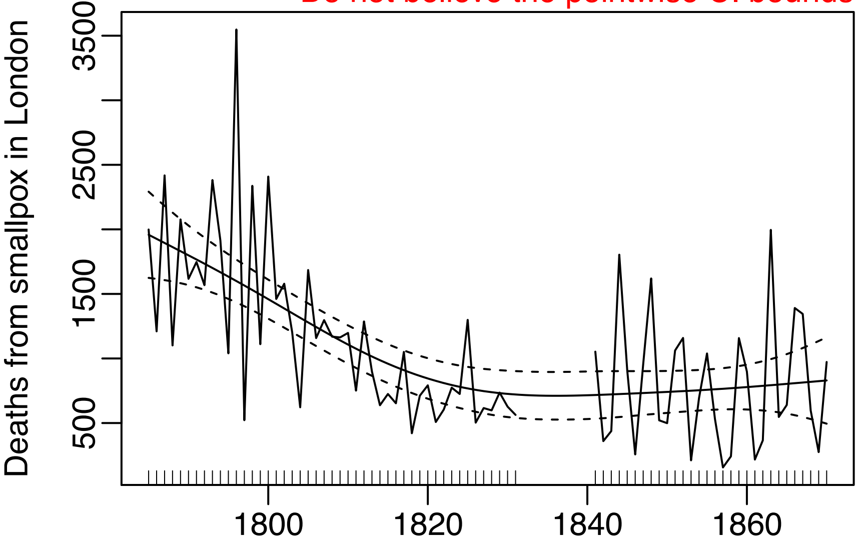
Estimation of an Underlying Trend Surface

from Diggle and Ribeiro 2007, p.151.

Estimation of a spatial trend $\mu(x) = \mathrm{E}[Y \mid x]$, where $Y(x)$ represents the reponse at location x, may be of interest [because]

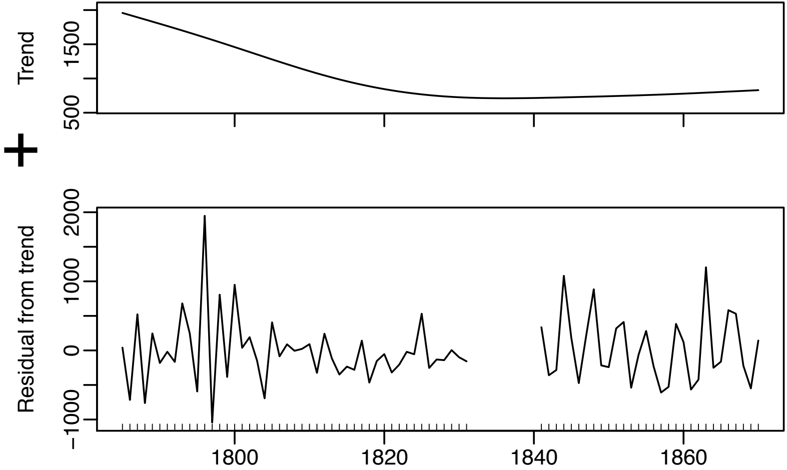
- . . . the experimenter has recorded a number of spatial explanatory variables, and wishes to to know which . . . influence the mean response.
- In other problems, . . . by including a term for the trend, typically modeled . . . as a low degree polynomial surface, we improve the precision of the resulting predictions.

Trend, in a time series



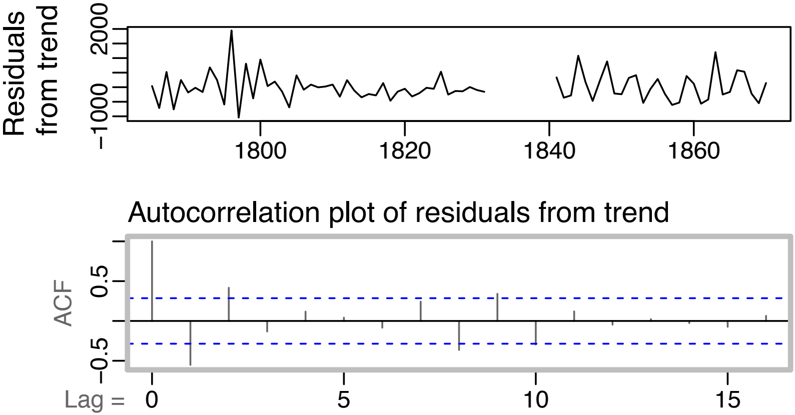
Do not believe the pointwise CI bounds!

Trend + Residual (then interpolate residual)



Correlation structure in the residuals may allow improvement on the smooth, for interpolating the missing years.

Use the Autocorrelation plot to look for pattern in residuals



In time series, autocovariance seems preferred to variogram.

Smoothing cannot accommodate the evident high/low pattern.

There may be variance heterogeneity.

The `meuse` dataset

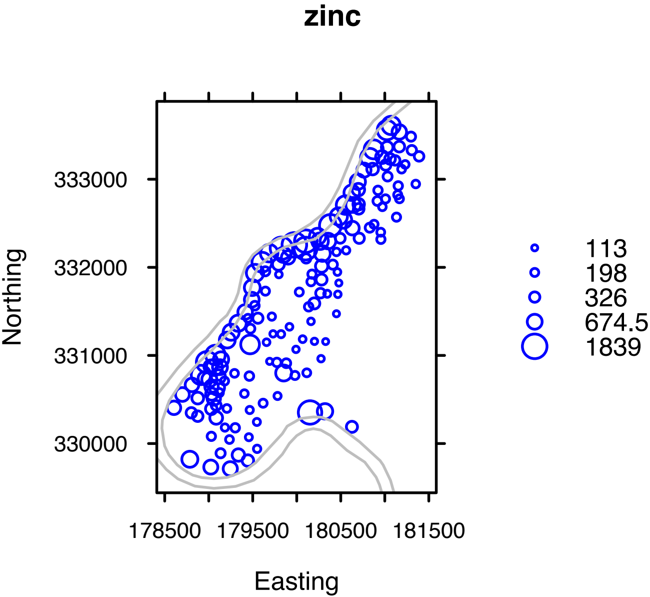
`dist`: distance from river

`elev`: elevation above river bed

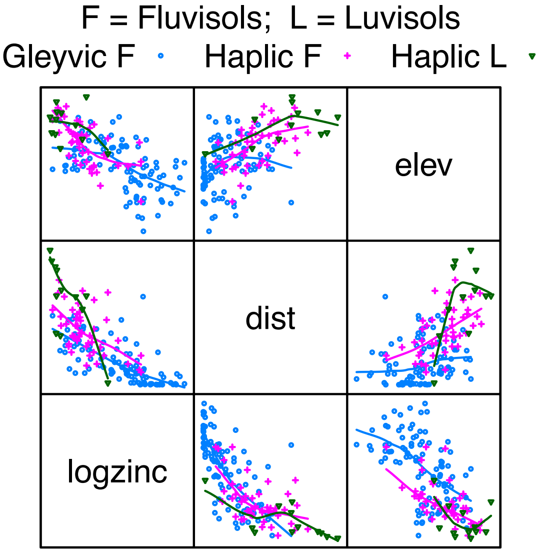
`soil`: soil type

`ffreq`: flooding frequency

Predict levels of `zinc` or other heavy metals.



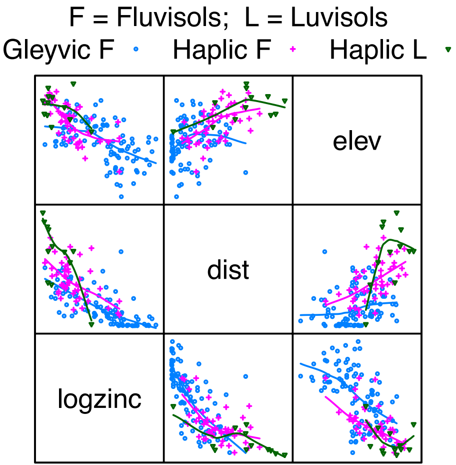
Graphics Tools – Scatterplot Matrix

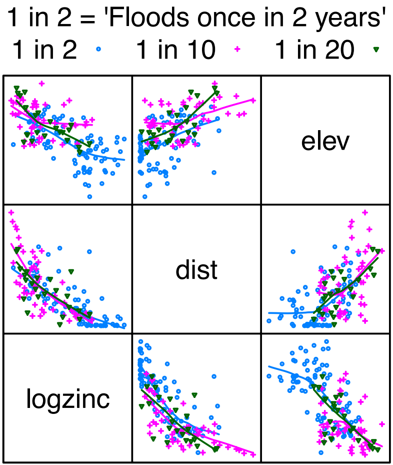


- GF: Calcareous soil
- HF: Non-calcareous
- HL: Red Brick

- For the variable on the vertical axis, look across the row to the diagonal
- For the variable on the horizontal axis, look up or down the column to the diagonal.

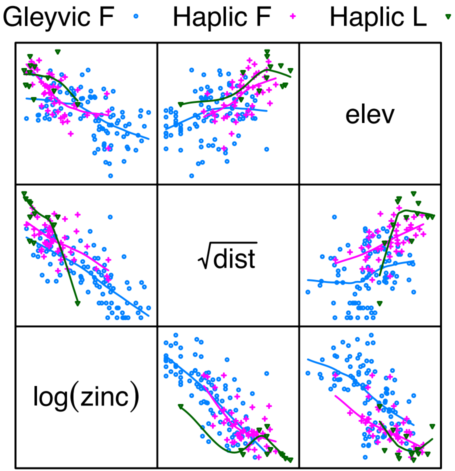
Scatterplot Matrix – by soil type and by flooding frequency

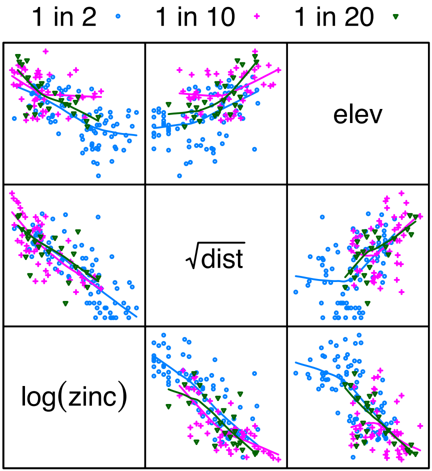




Relationships with `log(zinc)` differ a/c soil type and flooding frequency. Perhaps work with $\sqrt{\texttt{dist}}$?

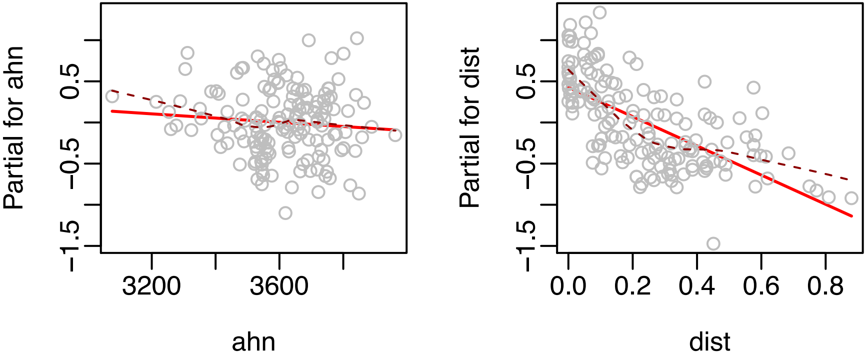
Scatterplot Matrix – work with square root of distance?





Termplots (detect non-linearity after model fitting)

```
lm.zinc <- lm(log(zinc) ~ ahn + dist + ffreq + soil,
              data=as.data.frame(meuse))
termplot(lm.zinc, terms=c("ahn","dist"),
         partial=TRUE, smooth=panel.smooth)
```



There is clear non-linearity, certainly for `dist`.

Model Simplification

```
> round(summary(lm.zinc)$coef, 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.269      1.137   6.391    0.000
ahn            0.000      0.000  -0.815    0.416
dist          -1.776      0.254  -7.004    0.000
ffreq.L       -0.357      0.087  -4.095    0.000
ffreq.Q        0.207      0.068   3.035    0.003
soil2         -0.444      0.095  -4.676    0.000
soil3         -0.074      0.162  -0.458    0.648
```

Drop `ahn` with almost no change to other coefficients – e.g., coefficient of `dist` changes from -1.882 to -1.776.

Model Simplification

```
> round(summary(lm.zinc)$coef, 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.269      1.137   6.391    0.000
ahn            0.000      0.000  -0.815    0.416
dist          -1.776      0.254  -7.004    0.000
ffreq.L       -0.357      0.087  -4.095    0.000
ffreq.Q        0.207      0.068   3.035    0.003
soil2         -0.444      0.095  -4.676    0.000
soil3         -0.074      0.162  -0.458    0.648
```

A Note on Principal Components: With effects as above, it will mix noise (`ahn`, `soil3`) with signal (other data)

From this point, use `elev` in place of `ahn`.

Strategy for Regression Modeling

1. Use scatterplot matrices to help decide whether/what transformations are indicated.
2. Think about what makes scientific sense.
3. Develop a strategy that will severely limit the nunber of models considered, e.g.
   - Model 1: Fit all main effects
   - Model 2: Main effects, plus all 2-term interactions
   - . . .

   Then choose the appropriate order of model.
4. Think model simplification, not variable selection.
   Or if prediction is the aim, think predictive accuracy!
5. Refine the model choice – maybe use `drop1()` & `add1()`.

Main effects model, vs model with 2-term interactions

```
## Model 1: Main effects only
form1 <- log(zinc) ~ rt2dist + elev + ffreq + soil
lm1.zinc <- lm(form1, data=as.data.frame(meuse))

## Model 2: Include also 2-term interactions
form2 <- log(zinc) ~ (rt2dist + elev + ffreq + soil)^2
lm2.zinc <- lm(form2, data=as.data.frame(meuse.ov))

> anova(lm1.zinc, lm2.zinc, test="F")
. . . .
        Res.Df    RSS Df Sum of Sq      F Pr(>F)
Model 1    148 20.475
Model 2    136 17.842 12    2.6325 1.6722 0.0795
```

…here, *p*-values are at best very rough indicators

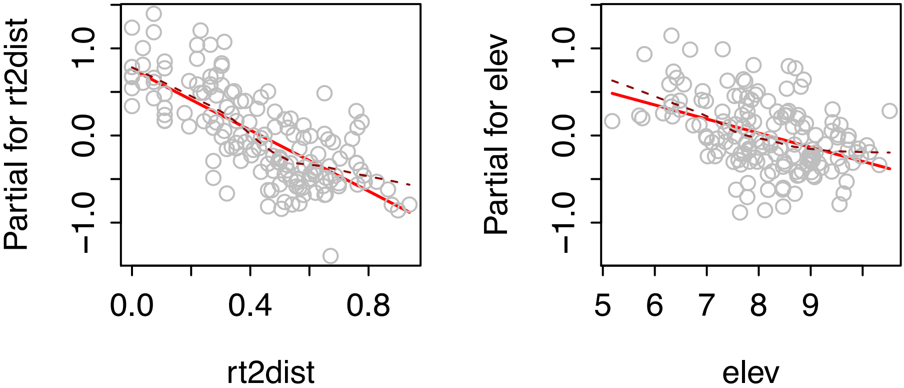
- Not testing hypotheses, but choosing models
- Nominal *p*-values can be very anti-conservative (too small)

Summary of estimates from main effects model

```
> round(summary(lm1.zinc)$coef,3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.968      0.321  24.798    0.000
rt2dist       -1.749      0.186  -9.388    0.000
elev          -0.162      0.040  -4.065    0.000
ffreq.L       -0.204      0.073  -2.796    0.006
ffreq.Q        0.118      0.062   1.897    0.060
soil2         -0.193      0.092  -2.112    0.036
soil3         -0.016      0.137  -0.118    0.906
```

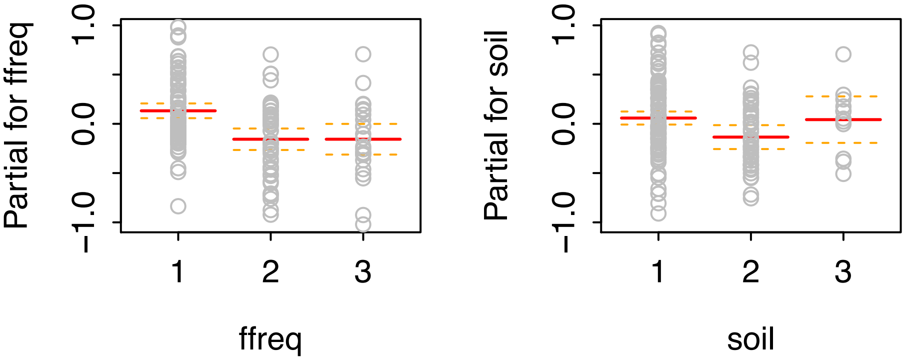
Check termplots for rt2dist and elev

```
termplot(lm1.zinc, terms=1:2, partial=TRUE,
         smooth=panel.smooth)
```



Check termplots for `soil` and `ffreq`

```
termplot(lm1.zinc, terms=c("ffreq","soil"),
         partial=TRUE, smooth=panel.smooth)
```



Treat the 95% confidence bounds with scepticism.

NB: As soil type 3 has only 12 points, combine types 1 & 3.

Cross-Validation (CV) accuracy will be our guide!

Basic idea – Training/Test, with a random data split

Do not use the test data for tuning or variable selection!
(this is a form of cheating!)

Cross-validation – a clever use of the training/test idea

- Split data into k parts (e.g., $k = 10$). Each division into training: 90% of data | test: 10% of data is a 'fold'.
- For each of the k possible 'folds', train on the training data, make predictions for the test data. At the end, predictions are available for all data.

Be sure to repeat any tuning and/or variable selection at each cross-validation "fold" (do not cheat!)

Cross-validation closely mimics practical prediction

CV: GAM fit, then apply Kriging to Residuals

1. Divide data into 10 parts.
2. For each fold, with the

training: 90% of data test: 10% of data split at that fold:

- Fit trend model to current training data, predict for test.
- Estimate variance parameters for residuals from the trend[1]
- Obtain kriged estimates for test data.

3. Calculate the CV estimate of residual mean square error:

(a) from the CV residuals from the trend surface

(b) from the CV residuals from the kriged estimates.

NB: Compare models across the same choice of folds.

[1] Here, assume a spherical variance structure

idw kriging

Variance of Mean square error?

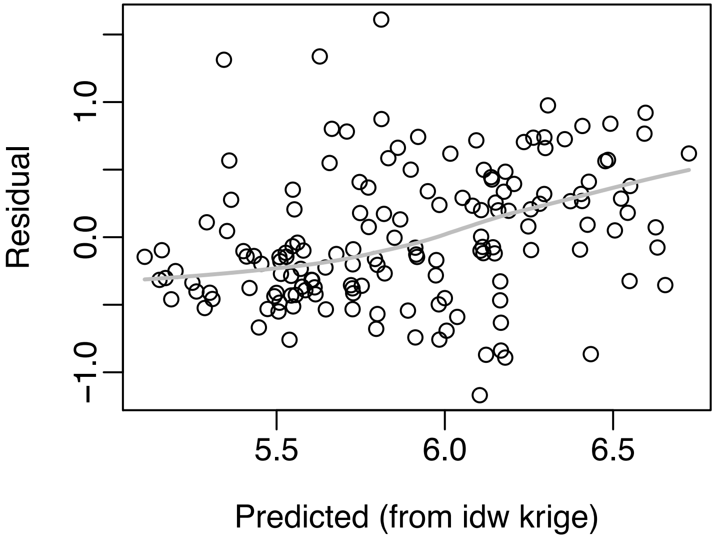
```
out <- as.data.frame(krige.cv(log(zinc)~1, meuse,
                nmax = 40, nfold=10)[c("residual","obs
css <- var(out$observed)
mse <- with(out, sum(residual^2)/length(residual))
resvar <-  with(out, var(residual))
print(round(c(mse=mse, r2=1-resvar/css, r2mse=1-mse/cs
```

Output:

```
  mse    r2 r2mse
0.235 0.546 0.549
```

Residuals vs predicted

```
plot(residual ~ I(observed-residual), data=out))
```



Nevertheless, mse and variance of residual are similar.

LM + kriging – 2-level interaction model vs main effects

Main effects only, vs model with all 2-term interactions

```
## Use JM’s function cv.gamkrige()
rhlm1 <- ~ rt2dist + Soil + elev + ffreq
cv.gamkrige(rhlm1, yvar="log(zinc)")
```

Mean square error (mse), from cross-validation

	Run 1		Run 2		Run 3	
	reg	reg+kr				
Main effects	0.146	0.096	0.142	0.093	0.149	0.102
2-level interact	0.157	0.117	0.150	0.107	0.162	0.121

Further refinement?

Add term `elev:ffreq` to the main effects model? The CV mean square error explained is however essentially unchanged.

Smoothing terms – automatic choice of smoothness

```
gam.smooth1 <-
  gam(log(zinc) ~ s(dist) + s(elev) + ffreq + Soil,
      data=as.data.frame(meuse))
```

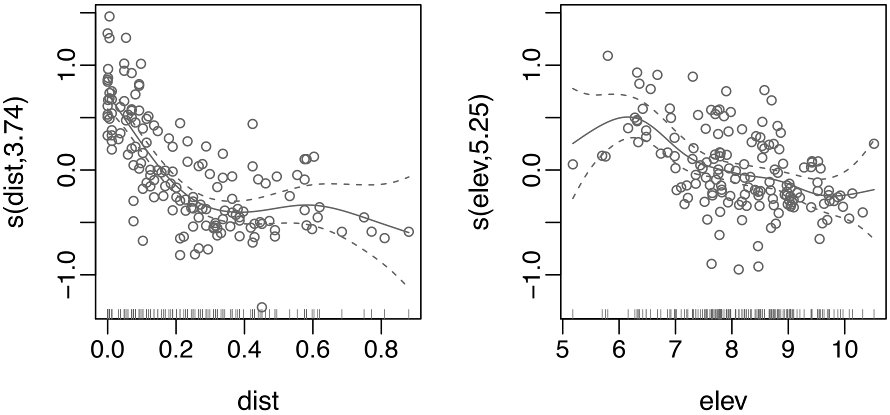
Selected Output from `summary(gam.smooth1)`

```
Approximate significance of smooth terms:
         edf Ref.df     F p-value
s(dist) 3.83   4.75 30.01 < 2e-16
s(elev) 5.13   6.28  5.21 5.3e-05

R-sq.(adj) =  0.776   Deviance explained = 79.4%
GCV score =  0.127  Scale est. = 0.117   n = 155
```

The *p*-values are much too small, & R-sq (adj) too large.

Termplots – `plot(gam.mod1, resid=TRUE, pch=1)`



$p < 2e - 16$ and $p = 0.000053$?

Formal ANOVA Model Comparisons

```
Models:
1: gam(log(zinc) ~ elev + rt2dist + ffreq + Soil
2:          ~ s(elev) + s(dist) + ffreq + Soil
3:          ~ s(elev, dist) + ffreq + Soil
4:          ~ s(elev, dist, by = ffreq) + ffreq + Soil
```

`anova()` **comparison:**

```
  Resid. Df Resid. Dev    Df Deviance    F  Pr(>F)
1       149       20.5
2       142       16.6  6.96     3.91 5.61 1.3e-05
3       134       15.1  8.00     1.52 1.90   0.066
4       122       12.2 12.35     2.86 2.31   0.010
```

The *p*-values are broad hints (even more than for lm models)!

- Theory is an (optimistic) approximation
- Observations are not independent.

GAM fit – alternative smooth term models

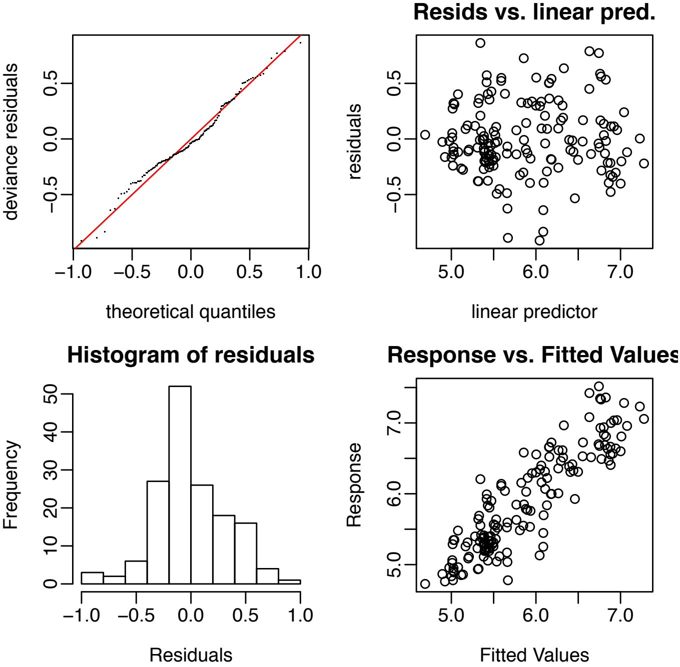
NB that the variance parameters are re-estimated at each fold.

```
## Smooths applied to main effects
gam1 <- log(zinc) ~ s(dist) + s(elev) + ffreq + Soil
gam.mod1 <- gam(gam1, data=meuse)

## Smooth surface
gam2 <- log(zinc) ~ s(dist, elev) + ffreq + Soil
. . . .

## One smooth surface for each level of ffreq
gam2ff <- log(zinc) ~ s(dist, elev, by=ffreq) +
  ffreq + Soil
. . . .
```

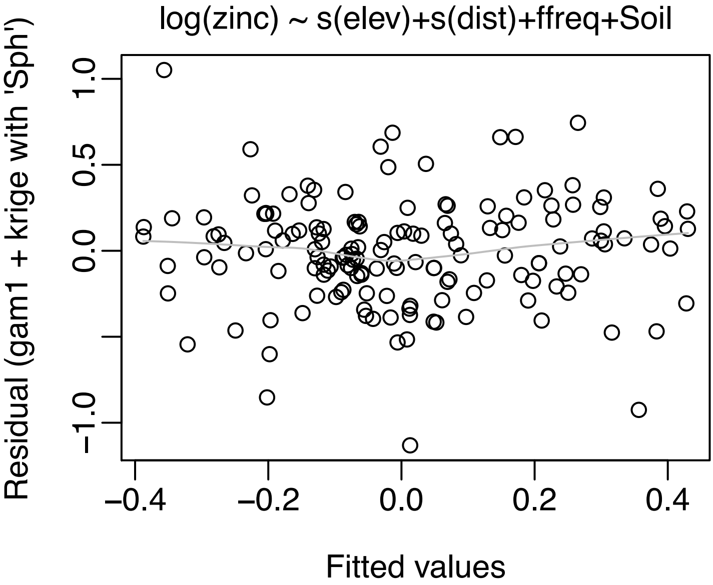
Kriging will then be applied to the residuals.



Diagnostics – smooths applied to main effects

```
gam1 <-
  log(zinc) ~
    s(dist) +
    s(elev) +
    ffreq +
    Soil
gam.mod1 <-
  gam(gam1,
      data=meuses
gam.check(
  gam.mod1)
```

Plot residuals from kriging vs fitted values



Summary of Results – CV R^2 vs Model adj R^2

	Run 1	Run 2	Run 3	Adj R^2
	reg reg+kr			(model)
Main effects	0.15 0.10	0.15 0.10	0.14 0.09	0.14
s(dist)+s(elev)	0.13 0.09	0.13 0.09	0.13 0.09	0.12
scale from CV	0.13 0.09	0.13 0.09	0.13 0.09	-
s(dist, elev)	0.15 0.11	0.14 0.11	0.14 0.11	0.12
scale from CV	0.15 0.10	0.15 0.10	0.14 0.10	-
add `by==ffreq`	0.16 0.13	0.15 0.12	0.15 0.12	0.15
scale from CV	0.17 0.12	0.15 0.11	0.15 0.12	-

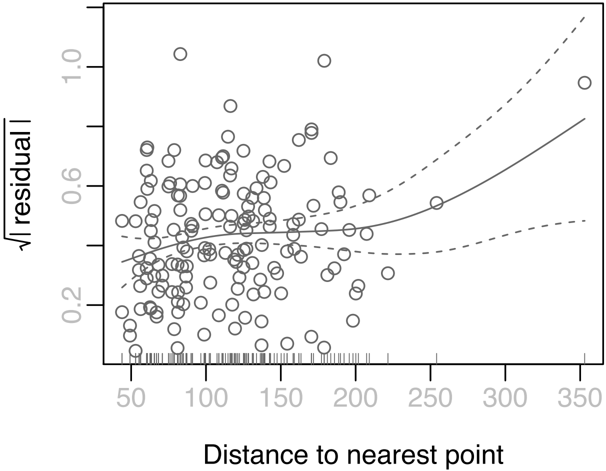
Replace kriging by smoothing on (x, y) coordinates?

```
cv.gamkrige(rhs =
            ~ s(elev) + s(dist) +
            ffreq + Soil +
            s(x,y),
            dokrige=FALSE)
```

CV mse $=$ 0.11 (residual ms^2 from model $=$ 0.070) i.e., scarcely different from kriging on residuals.

Would smoothing on (x, y) coordinates as above, then kriging on residuals do better still?
Here, it seems not. The semi-variogram from the residuals is flat.

How does accuracy changes with distance to nearest point?



1. For each test point, calculate distance to nearest training point.

2. Plot $\sqrt{(abs(\texttt{residual}))}$ versus this distance.

Results (mean square error) with `ahn` in place of `elev`

	reg	reg+kr
Main effects	0.14	0.10
s(dist)+s(ahn)	0.13	0.09
s(dist, ahn)	0.14	0.11
`add by==ffreq`	0.15	0.12

Comments and Questions

- Without modification `gam()` gives a trend surface that is almost certainly an over-fit. Using the CV result to set the `gam()` scale parameter seemed a slight improvement.
- Addition of a smooth on the coordinate axes to the trend surface did almost as well as kriging on residuals from the trend surface. The smallpox time series suggests that the kriging approach should in certain circumstances do better. Laslett (1994) has examples that support this.
- A better appraoch might be to modify `krige()` to allow smooth trend terms, using cross-validation to optimise over the choice of roughness penalty.

Code Used

Give access to required packages

```
library(mgcv)
library(gstat)
```

Give access to or create the data object `meusesp`

Go to `http://maths.anu.edu.au/~johnm/r/spatial`

Load the data object `meusesp.RData`.

(for creating `meusesp` anew, see the **README.txt** file)

Try out functions that will do the calculations

```
cv.gamkrige()
compare.gamkrige()
```

Further Reading

Diggle, Peter J & Ribeiro Jr, Paulo J 2007. Model-Based Geostatistics. Springer.

Laslett, Geoffrey M 1994: Kriging and Splines: An Empirical Comparison of Their Predictive Performance in Some Applications. JASA 89, 391-400

See also references given in the Laslett paper, notably to Grace Wahba's work.

Acknowledgement

Thanks to Jin Li (Geosciences Australia) for helpful comments.