
Applied geostatistics

Exercise 6

Assessing the quality of spatial predictions

Geostatistical simulation

D G Rossiter
University of Twente, Faculty of Geo-Information Science & Earth
Observation (ITC)

January 6, 2014

Contents

1	Introduction	1
2	Model evaluation	1
2.1	Actual vs. predicted values: 1:1 line	1
2.2	* Predicted vs. actual values: linear regression	4
2.3	Answers	9
3	Cross-validation of geostatistical models	9
3.1	* LOOCV and the variogram	15
3.2	Answers	18
4	Kriging prediction variance	20
4.1	Answers	21
5	Conditional simulation	21
5.1	Answers	26
6	* Unconditional simulation	27
6.1	Answers	35
7	Self-test	37

Version 2.3 Copyright © 2007–2009 ITC, 2010, 2012–2014 University of Twente All rights reserved. Reproduction and dissemination of the work as a whole (not parts) freely permitted if this original copyright notice is included. Sale or placement on a web site where payment must be made to access this document is strictly prohibited. To adapt or translate please contact the author (<http://www.itc.nl/personal/rossiter>).

References	38
Index of R concepts	39

1 Introduction

After completing this exercise you should be able to:

1. Evaluate model predictions with an independent dataset;
2. Evaluate geostatistical models with cross-validation;
3. Simulate random fields as realizations of spatial structure, both unconditionally and conditioned on known points.

2 Model evaluation

This section is a continuation of Exercise 4, §4.2. Recall that we used 259 of the observations (the **calibration** dataset) to **model** the spatial structure (i.e. fit a variogram model) and to **predict** at 100 points where we had also sampled (the **evaluation**, sometimes called **validation** dataset).

Since we didn't use the 100 points of the evaluation dataset either to make the model or to predict, these are an **independent** test of the model. We can compare the predictions to the actual values.

Task 1 : Locate your results from Exercise 4, §4.2:

1. Validation points `jura.val`
2. Calibration points `jura.cal`
3. Fitted variogram model `vmf`
4. Predictions `k.val`

•

There are two ways to compare actual vs. predicted values:

- Against a 1:1 line (§2.1)
- With a linear regression of actual vs. predicted (§2.2)

We now examine these in turn.

2.1 Actual vs. predicted values: 1:1 line

Now we compare the **predictions** with the **actual values** at the 100 evaluation points. Conceptually, we are comparing them on a 1:1 line: intercept is 0 (no **bias**) and the slope is set at 1 (**gain** is equal).

Task 2 : Compare the predictions to the actual values: with a numerical summary, a histogram of the differences, and a bubble plot of the differences.

•

Note: In regression diagnostics it is customary to subtract the predictions from the actual values; however we want to express this as **prediction errors**, so we reverse the sign: predicted less actual.

We use the `summary` method for the numeric summaries; the `hist` method for the histogram; the `bubble` method for the bubble plot. This last requires a spatial object, so we first convert the prediction errors to a (one-field) data frame with the `as.data.frame` method and then specify its coordinates with the `coordinates` method; note that one form of this extracts the coordinates of an existing spatial object (here, `jura.val`) and another form specifies the coordinates of a new spatial object (here, `pred.error`).

```
> summary(k.val$var1.pred); summary(jura.val$Co)
```

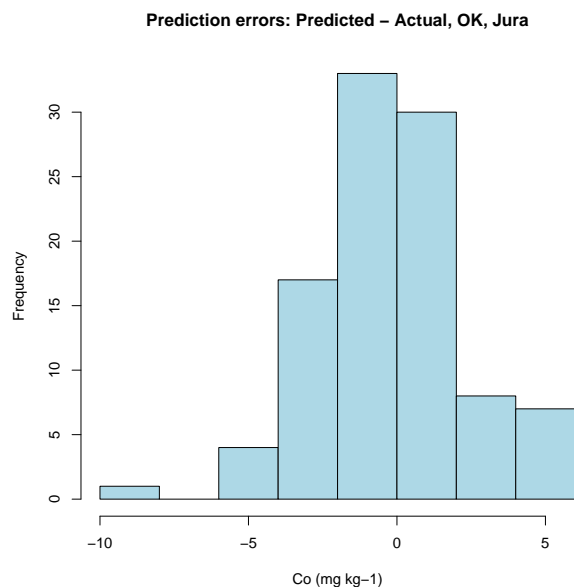
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.58	7.70	10.10	9.46	11.30	14.00

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.65	7.95	10.10	9.79	12.50	20.60

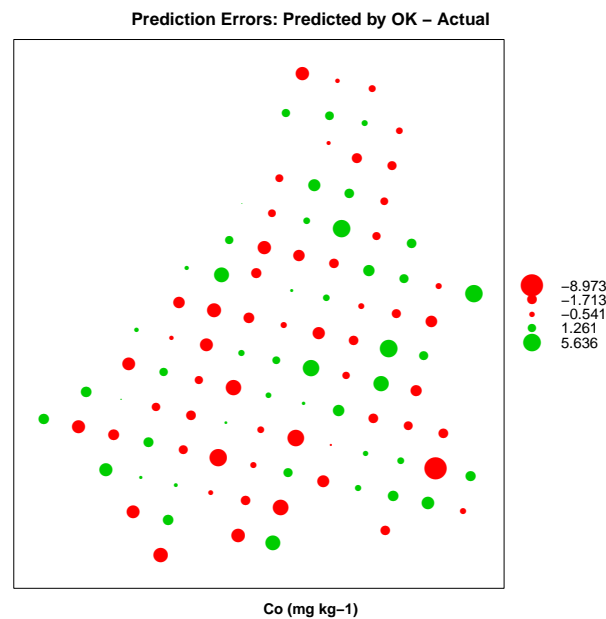
```
> pred.error <- k.val$var1.pred - jura.val$Co
> summary(pred.error)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-8.970	-1.710	-0.541	-0.331	1.260	5.640

```
> hist(pred.error,
+       main="Prediction errors: Predicted - Actual, OK, Jura",
+       xlab="Co (mg kg-1)", col="lightblue")
> pred.error <- as.data.frame(pred.error)
> coordinates(pred.error) <- coordinates(jura.val)
```



```
> print(bubble(pred.error,
+             main="Prediction Errors: Predicted by OK - Actual",
+             sub="Co (mg kg-1)"))
```



Q1 : *Is there any spatial pattern to the evaluation prediction errors?* [Jump to A1](#) •

Task 3 : Compute the **Mean Prediction Error** (MPE), also called the **bias**, and the **Root Mean Squared Error** (RMSE), also called the **precision**. Compare to the mean and standard deviation of the target variable in the calibration dataset; these are the **relative** MPE and RMSE. •

We can also compare to the **inter-quartile range** (the range from the 25% to the 75% quantiles of the original variable, i.e. with the central half of the values), with the IQR method.

```
> summary(k.val$var1.pred - jura.val$Co)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-8.970 -1.710  -0.541  -0.331  1.260   5.640

> (mpe <- mean(k.val$var1.pred - jura.val$Co))

[1] -0.33123

> (rmse <- sqrt(mean((jura.val$Co - k.val$var1.pred)^2)))

[1] 2.487

> summary(jura.cal$Co)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.55   6.52   9.76   9.30  12.00  17.70

> mpe/mean(jura.cal$Co)

[1] -0.035612
```

```
> rmse/sd(jura.cal$Co)

[1] 0.69524

> rmse/IQR(jura.cal$Co)

[1] 0.45549
```

Q2 : *What are the MPE (bias) and RMSE (precision)? How do these compare with the range of the target variable?* Jump to A2 •

2.2 * Predicted vs. actual values: linear regression

Another way to evaluate the prediction is with a **linear regression** between actual and predicted values. Ideally, this would be a 1:1 line: intercept is 0 (no **bias**) and the slope is set at 1 (**gain** is equal). In the previous §2.1 we assume that; here we test it.

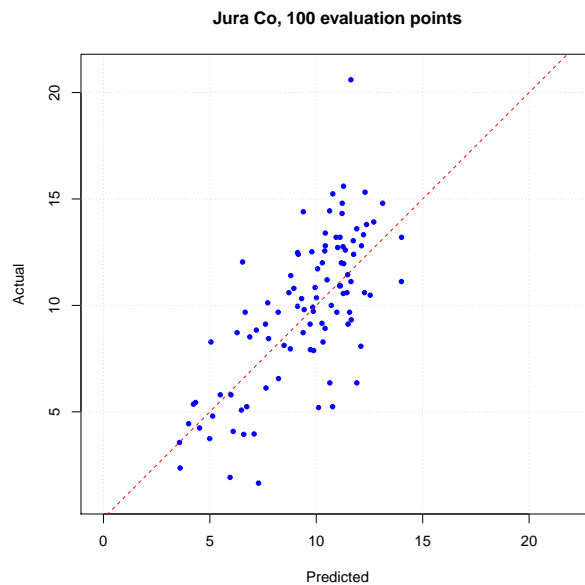
The bivariate regression of two variables on each other depends on which variables is considered the predictor and which the predictand. Here it makes sense to use the actual (true) values as the **dependent** or **predictand** variable and the predicted values as the **independent** or **predictor** variable; the question is how well does the fitted model predict the actual? This is the usual order in agricultural simulation models [e.g. 1].

Note: This is a bit confusing: the predictor in this regression, i.e. the independent variable, is the predicted value from the model, and the predictand is the actual value from observation. Thus the bias and gain refer to the performance of the model in trying to predict the actual. For example, a positive bias means that the actual is systematically higher than the model; the model thus under-predicts.

Task 4 : Display a square feature-space scatterplot of the actual (abscissa) vs. predicted (ordinate) Co concentrations at the 100 evaluation points, with a 1:1 line superimposed. •

Note that the `asp=1` argument to the `plot.xy` method makes the plot square. To determine the axis limits, we use the `min` and `max` methods to determine the minimum and maximum data values of both sets; then the `floor` and `ceiling` (“ceiling”) methods to reach the next lowest and highest integers.

```
> plot.lims=c(floor(min(jura.val$Co, k.val$var1.pred)),
+ ceiling(max(jura.val$Co, k.val$var1.pred)))
> plot(jura.val$Co ~ k.val$var1.pred, asp=1, pch=20,
+      col="blue", ylab="Actual", xlab="Predicted",
+      xlim=plot.lims,
+      ylim=plot.lims, main="Jura Co, 100 evaluation points")
> abline(0,1, lty=2, col="red")
> grid()
```



Q3 : Does the scatterplot appear to be well-fitted by the 1:1 line? [Jump to A3](#) •

Task 5 : Compute the linear regression of predicted cobalt concentrations vs. the actual values, at the 100 evaluation locations. Summarize the model and plot the regression diagnostics: fitted vs. residuals, and normal quantile plot of the residuals. •

```
> lm.val <- lm(jura.val$Co ~ k.val$var1.pred)
> summary(lm.val)

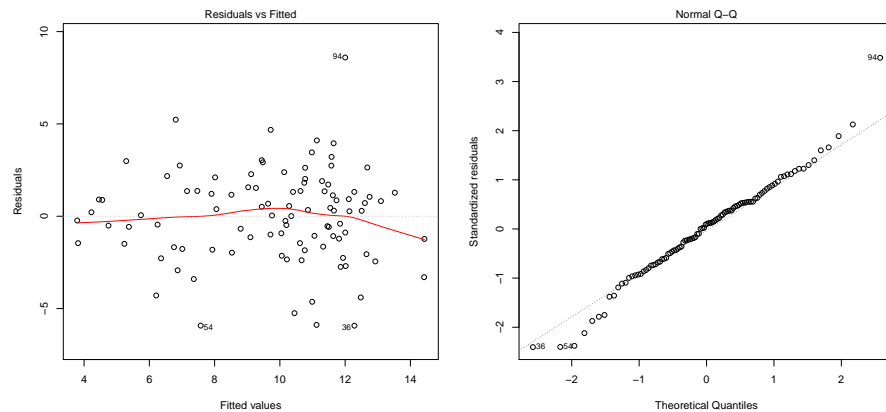
Call:
lm(formula = jura.val$Co ~ k.val$var1.pred)

Residuals:
    Min       1Q   Median       3Q      Max
-5.925 -1.533  0.241  1.368  8.598

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.141     0.986    0.14   0.89
k.val$var1.pred 1.020     0.101   10.12 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.49 on 98 degrees of freedom
Multiple R-squared:  0.511,    Adjusted R-squared:  0.506
F-statistic: 102 on 1 and 98 DF,  p-value: <2e-16

> par(mfrow = c(1, 2))
> plot(lm.val, which = 1:2)
> par(mfrow = c(1, 1))
```



Q4 : *What is the regression equation?*

Jump to A4 •

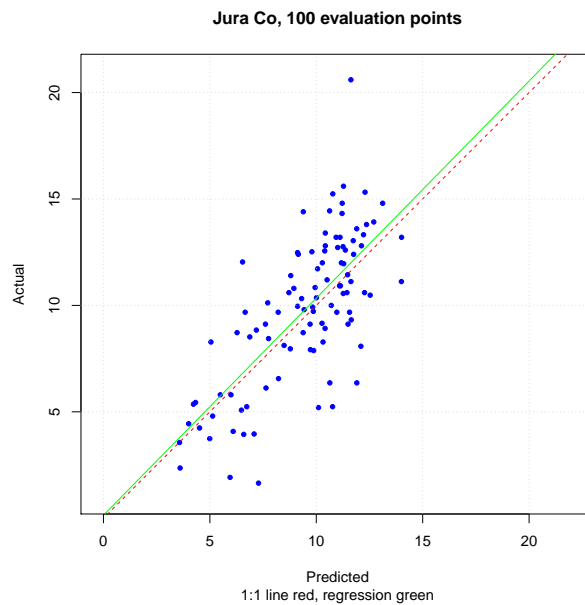
The **bias** is the intercept, and the **gain** is the slope.

Q5 : *What is the bias and gain? What should these values be?* *Jump to A5 •*

Q6 : *Does a linear relation appear to be justified from the diagnostics?* *Jump to A6 •*

Task 6 : Plot the regression line of predicted vs. actual on the scatterplot, also with the 1:1 line. •

```
> plot(jura.val$Co ~ k.val$var1.pred, asp=1, pch=20,
+      col="blue", xlab="Predicted", ylab="Actual",
+      xlim=plot.lims, ylim=plot.lims,
+      main="Jura Co, 100 evaluation points",
+      sub="1:1 line red, regression green")
> abline(0,1, lty=2, col="red")
> abline(lm.val, col="green"); grid()
```

We can test the bias against the null hypothesis of no bias, directly from the t-test of the corresponding coefficient.

Q7 : *What is the probability that rejecting the null hypothesis of no bias is a Type I error?* Jump to A7 •

The t-test for the slope is against zero slope, not one; so we test this seeing if 1 is within a confidence interval for the slope.

Task 7 : Compute the 90% confidence interval for the slope of the evaluation regression. •

The `summary.lm` method, called by the generic `summary` method for objects of type `lm`, summarizes the model; in particular it computes the coefficients and their standard errors. The `coef` generic method, applied to the results of `summary.lm`. We can see its structure:

```
> coef(summary(lm.val))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.14088	0.98581	0.14291	8.8665e-01
k.val\$var1.pred	1.02012	0.10082	10.11848	6.6956e-17

From this it is clear that the second row is the slope; the first column is the estimated coefficient, and the second is the standard error. The confidence interval uses the t-value for the residual degrees of freedom, which is the `df` field of the model object:

```
> lm.val$df
```

```
[1] 98
```

This is just the number of data values less the number of parameters.

We use the `qt` method to extract the t-value:

```
> qt(0.95, lm.val$df)

[1] 1.6606
```

Note that this is a two-sided confidence interval, so we use half of the interval for each tail. We have fixed the confidence level $\alpha = 0.10$, so the required quantile of the t-distribution is $(1 - \alpha/2) = 0.95$

With all this information, we can now compute the upper and lower confidence intervals for the slope:

```
> t.val <- qt(0.95, lm.val$df)
> coef(summary(lm.val))[2, 1] - t.val * coef(summary(lm.val))[2,
+      2]

[1] 0.85271

> coef(summary(lm.val))[2, 1] + t.val * coef(summary(lm.val))[2,
+      2]

[1] 1.1875
```

Q8 : *Does the computed confidence interval include 1? Should we reject the null hypothesis of no gain?* *Jump to A8 •*

In summary, this regression shows no evidence of bias or gain in the kriging predictions at evaluation points. So, we can quantify how close the predictions come to reality.

Q9 : *Overall, how close are the model predictions to the actual values?* *Jump to A9 •*

Once we've established a linear relation, the goodness-of-fit can be measured with the coefficient of determination; this is given by the `adj.r.squared` field of the model summary:

```
> summary(lm.val)$adj.r.squared

[1] 0.50595
```

Task 8 : Clean up from this section. •

```
> rm(pred.error, rmse, mpe, plot.lims, lm.val, t.val)
```

2.3 Answers

A1 : No, both positive and negative evaluation prediction errors are found everywhere, with no clustering; large and small prediction errors are likewise not clustered. Thus there are no parts of the study area where OK is systematically better or worse. [Return to Q1](#) •

A2 : The MPE (bias) is $-0.33 \text{ mg kg}^{-1} \text{ Co}$; this is a small proportion of the mean: -3.6% relative error.

The RMSE (precision) is $2.49 \text{ mg kg}^{-1} \text{ Co}$; this is less than half of (0.46 times) the IQR: $(12.00 - 6.52) = 5.48$ and about two-thirds (0.70 times) the sample set standard deviation. Thus the evaluation precision is higher than in the sample set. [Return to Q2](#) •

A3 : By eye it does not look 1:1; in particular, it appears that high actual values are substantially under-predicted. But we will see this is not the case; so if you thought it looked 1:1, your eye is better than mine. [Return to Q3](#) •

A4 : Actual value = $0.141 + 1.02 \cdot \text{Predicted value}$; this should be Actual value = Predicted value. [Return to Q4](#) •

A5 : The bias is the intercept: 0.141; it should be zero. The gain is the slope: 1.02; it should be one. [Return to Q5](#) •

A6 : There is one very poorly-modelled point (number 94); otherwise the QQ-plot supports a normal distribution of the residuals, and there is no evidence of heteroscedascity. [Return to Q6](#) •

A7 : The probability that rejecting the null hypothesis of no bias is a Type I error is given by the $\text{Pr}(>|\mathbf{t}|)$ column of the model summary, for the intercept; it is 0.887. Thus the null hypothesis of no bias should not be rejected; the estimates of 0.141 is likely different from zero only by chance. [Return to Q7](#) •

A8 : The 90% confidence interval includes 1, so we can not reject the null hypothesis of no gain, without running at least a 10% risk of being wrong. [Return to Q8](#) •

A9 : The model explains only 50.6 % of the variation in actual values. This is not very precise. [Return to Q9](#) •

3 Cross-validation of geostatistical models

With a geostatistical prediction, there are two steps:

1. Modelling the spatial structure; e.g. **estimating** the variogram parameters;
2. **Predicting** at unsampled locations.

These can be separate processes; for example, the variogram may be known from previous studies. Even if the same dataset is used for modelling and predicting, each single point has little influence on the **model** (recall the large number of point-pairs). So, it is possible to use the model to predict **at each observation point** separately, using all the other observations. This is called **leave-one-out cross-validation** (LOOCV).

Note: The observed value is left out, because kriging is an exact predictor at known points, and so would predict the value itself, if that point were included.

The `gstat` package supplies a `krige.cv` method for this.

Task 9 : Compute the leave-one-out cross-validation of OK of the Co value, for the **calibration** dataset. •

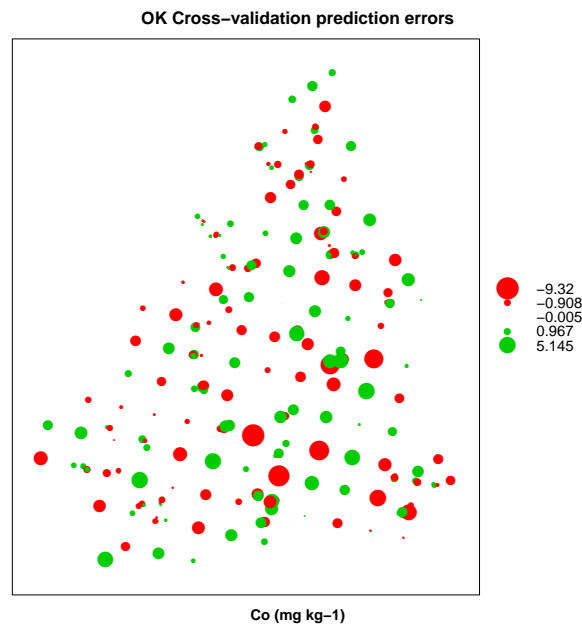
```
> k.cv <- krige.cv(Co ~ 1, loc = jura.cal, model = vmf)

[using ordinary kriging]
[using ordinary kriging]
...
```

Task 10 : Make a bubble plot of the residuals. •

Object `k.cv` is already spatial, with coördinates:

```
> print(bubble(k.cv, zcol="residual",
+             main="OK Cross-validation prediction errors",
+             sub="Co (mg kg-1)"))
```



Q10 : *Is there any spatial pattern to the cross-validation prediction errors?*
Jump to A10 •

Task 11 : Compute the absolute and relative MPE and RMSE of the cross-validation. •

The `krige.cv` method also computes the prediction errors; the MPE is their mean, and the RMSE is the square root of the sum of squared prediction errors:

```
> summary(k.cv)
```

Object of class SpatialPointsDataFrame
Coordinates:
 min max
X 0.626 4.92
Y 0.580 5.69
Is projected: NA
proj4string : [NA]
Number of points: 259
Data attributes:

var1.pred		var1.var		observed		residual	
Min.	: 3.24	Min.	: 1.93	Min.	: 1.55	Min.	: -9.3202
1st Qu.	: 7.48	1st Qu.	: 2.08	1st Qu.	: 6.52	1st Qu.	: -0.9077
Median	: 9.61	Median	: 2.66	Median	: 9.76	Median	: -0.0046
Mean	: 9.37	Mean	: 3.68	Mean	: 9.30	Mean	: -0.0737
3rd Qu.	: 11.58	3rd Qu.	: 5.62	3rd Qu.	: 11.98	3rd Qu.	: 0.9670
Max.	: 15.98	Max.	: 8.51	Max.	: 17.72	Max.	: 5.1446

zscore		fold	
Min.	: -4.2139	Min.	: 1.0
1st Qu.	: -0.5824	1st Qu.	: 65.5
Median	: -0.0019	Median	: 130.0

```

Mean      :-0.0203   Mean      :130.0
3rd Qu.: 0.5315   3rd Qu.:194.5
Max.      : 2.8312   Max.      :259.0

> (mpe <- mean(k.cv$residual))

[1] -0.073707

> mpe/mean(jura.cal$Co)

[1] -0.0079247

> (rmse <- sqrt(mean(k.cv$residual^2)))

[1] 2.0904

> rmse/sd(jura.cal$Co)

[1] 0.58438

> rmse/IQR(jura.cal$Co)

[1] 0.38286

```

Q11 : What are the absolute and relative (to the mean) MPE (bias) of this OK? What are the maximum errors? What are the absolute and relative (to the standard deviation) RMSE (precision)? *Jump to A11 •*

Another measure of model quality is the Mean Squared Deviation Ratio (MSDR) of prediction errors with kriging variance:

$$\text{MSDR} = \frac{1}{n} \sum_{i=1}^n \frac{\{z(\mathbf{x}_i) - \hat{z}(\mathbf{x}_i)\}^2}{\hat{\sigma}^2(\mathbf{x}_i)}$$

where $\hat{\sigma}^2(\mathbf{x}_i)$ is the kriging variance at cross-validation point \mathbf{x}_i , obtained during the kriging procedure (not the cross-validation).

The MSDR is a measure of the variability of the cross-validation vs. the variability of the sample set. This ratio should be 1. If it's higher, the kriging prediction was too optimistic about the variability.

Task 12 : Compute the MSDR of the cross-validation. •

The squared deviations were stored in the kriging prediction object `k.cv` as field `residual`; the kriging variance as field `var1.var`, so the MSDR is just the mean of the pairwise ratios of these two vectors:

```

> mean(k.cv$residual^2/k.cv$var1.var)

[1] 1.1117

```

Q12 : What is the MSDR? Is the kriging prediction more or less variable than the actual dataset? Is this a large difference? *Jump to A12 •*

Q13 : What would happen to the MSDR if we removed the nugget component from the model, i.e. specified a nugget of zero? [Jump to A13](#)

•

Task 13 : Remove temporary objects from the workspace; leave the `k.cv` cross-validation kriging predictions, they will be used in the next section. •

```
> rm(mpe, rmse)
```

Task 14 : **Optional:** Repeat the analysis of §2.2 (evaluation by linear regression) on the cross-validation predictions. •

First, the regression model, compared on a scatterplot with the 1:1 line:

```
> summary(lm.cv <- lm(jura.cal$Co ~ k.cv$var1.pred))

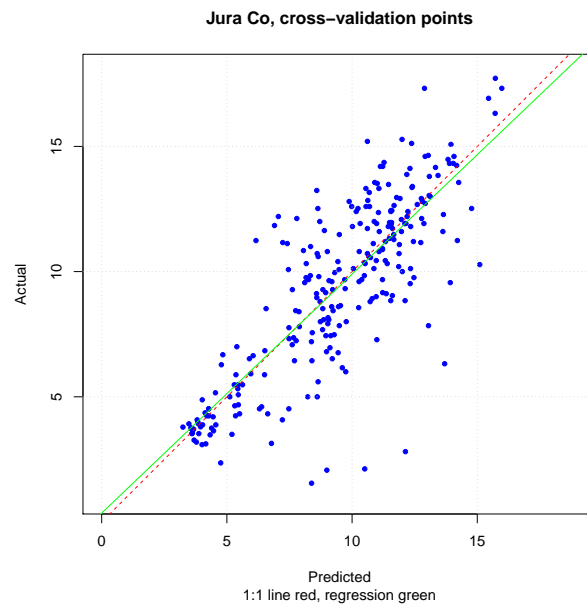
Call:
lm(formula = jura.cal$Co ~ k.cv$var1.pred)

Residuals:
    Min       1Q   Median       3Q      Max
-9.120 -0.924  0.004  1.156  5.112

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.3553     0.4218   0.84    0.4
k.cv$var1.pred  0.9542     0.0428  22.29 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

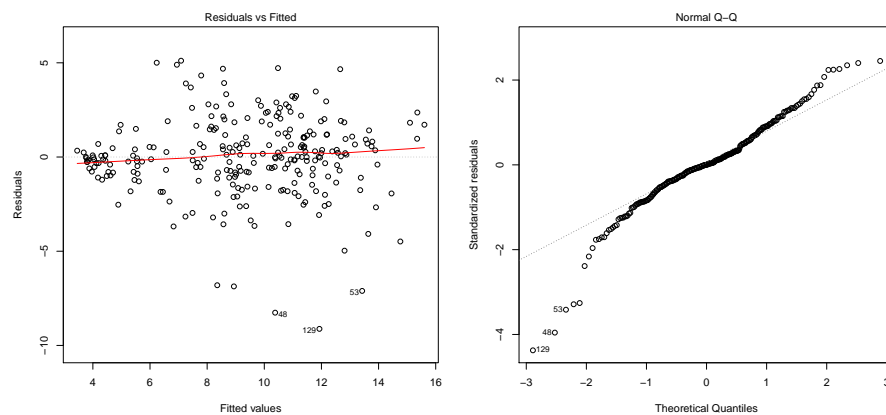
Residual standard error: 2.09 on 257 degrees of freedom
Multiple R-squared:  0.659,    Adjusted R-squared:  0.658
F-statistic: 497 on 1 and 257 DF,  p-value: <2e-16

> plot.lims=c(floor(min(jura.cal$Co, k.cv$var1.pred)),
+ ceiling(max(jura.cal$Co, k.cv$var1.pred)))
> plot(jura.cal$Co ~ k.cv$var1.pred, asp=1, pch=20,
+      col="blue", xlab="Predicted", ylab="Actual",
+      xlim=plot.lims, ylim=plot.lims,
+      main="Jura Co, cross-validation points",
+      sub="1:1 line red, regression green")
> abline(0,1, lty=2, col="red")
> abline(lm.cv, col="green"); grid()
```



Second, the regression diagnostic plots:

```
> par(mfrow = c(1, 2))
> plot(lm.cv, which = 1:2)
> par(mfrow = c(1, 1))
```



The confidence interval on the slope:

```
> t.val <- qt(0.95, lm.cv$df)
> coef(summary(lm.cv))[2, 1] - t.val * coef(summary(lm.cv))[2,
+ 2]

[1] 0.88357

> coef(summary(lm.cv))[2, 1] + t.val * coef(summary(lm.cv))[2,
+ 2]

[1] 1.0249

> rm(t.val)
```


The proportion of variability explained:

```
> summary(lm.cv)$adj.r.squared  
[1] 0.65779
```

Repeating the questions of §2.2:

Q14 : *What is the bias and gain?* *Jump to A14 •*

Q15 : *Does a linear relation appear to be justified from the diagnostics?* *Jump to A15 •*

Q16 : *What is the probability that rejecting the null hypothesis of no bias is a Type I error?* *Jump to A16 •*

Q17 : *Does the computed confidence interval include 1? Should we reject the null hypothesis of no gain?* *Jump to A17 •*

Q18 : *Overall, how close are the model predictions to the actual values?* *Jump to A18 •*

```
> rm(lm.cv)
```

3.1 * LOOCV and the variogram

In this **optional** section we show that omitting one point does not substantially affect the fitted variogram model. This implies that LOOCV with a single variogram model, estimated from all points, is valid. We investigate this heuristically with the Jura calibration dataset, by removing the highest value (i.e., the one that should have the largest influence on the variogram), and seeing what happens to the variogram estimation, variogram model fitting, and cross-validation prediction.

Task 15 : Identify the point in the calibration data set with the highest Co concentration. •

```
> jura.cal[ix <- which.max(jura.cal$Co), ]  
  
      coordinates      Rock  Land  Cd   Cu  Pb   Co   Cr  Ni  
225 (4.412, 1.088) Quaternary Meadow 1.53 17.72 47.6 17.72 39.52 26.4  
      Zn  
225 80.8
```

Task 16 : Compute the empirical variogram with and without this point, to the 1.6 km cutoff. •

```
> v <- variogram(Co ~ 1, loc = jura.cal, cutoff = 1.6)
> vv <- variogram(Co ~ 1, loc = jura.cal[-ix, ], cutoff = 1.6)
```

Q19 : *How many and what proportion of point-pairs have been removed from the empirical variogram?* *Jump to A19* •

```
> (sv <- sum(v$np))
[1] 12632
> (svv <- sum(vv$np))
[1] 12572
> print(sv - svv)
[1] 60
> print(1 - svv/sv)
[1] 0.0047498
```

Task 17 : Fit a model to this empirical variogram, beginning with the model found when using all point-pairs. Compare the fitted parameters to those for the all point-pair model. •

```
> (vvmf <- fit.variogram(vv, vmf))
      model  psill  range
1   Nug  1.294 0.0000
2   Pen 12.960 1.5322
> (diff.psill <- vvmf[2, "psill"] - vmf[2, "psill"])
[1] 0.028175
> (diff.range <- vvmf[2, "range"] - vmf[2, "range"])
[1] 0.0082585
> (diff.nugget <- vvmf[1, "psill"] - vmf[1, "psill"])
[1] -0.077183
> ((diff.psill/vmf[2, "psill"]) * 100)
[1] 0.21787
> ((diff.range/vmf[2, "range"]) * 100)
[1] 0.54193
```

```
> ((diff.nugget/vmf[1, "psill"]) * 100)

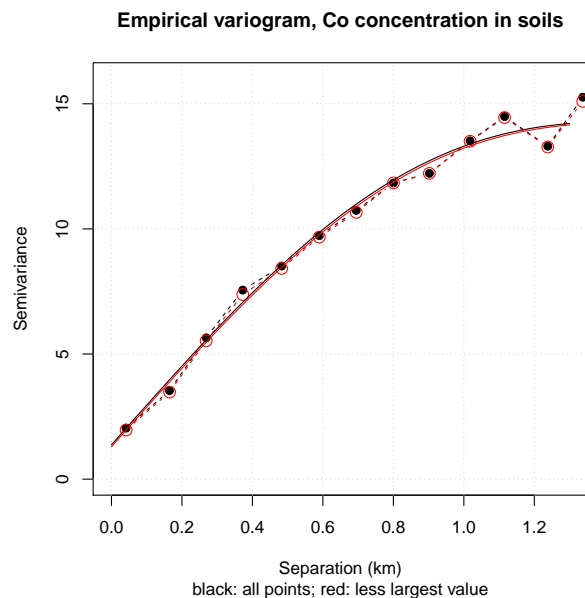
[1] -5.6289
```

Q20 : *How much did the variogram model parameters change?* [Jump to A20](#) •

Task 18 : Display both empirical variograms and both models on the same figure. •

We use the base graphics `plot` function, along with `title`, `points`, `lines` and `grid` as with any scatterplot. The only trick here is the use of the `variogramLine` function of `gstat` to compute semivariance values for the fitted models.

```
> plot(v$gamma ~ v$dist, xlim = c(0, 1.3), ylim = c(0,
+   16), pch = 20, cex = 1.5, xlab = "Separation (km)",
+   ylab = "Semivariance", type = "b", lty = 2)
> title(main = "Empirical variogram, Co concentration in soils",
+   sub = "black: all points; red: less largest value")
> points(vv$gamma ~ vv$dist, col = "red", pch = 1, cex = 1.5,
+   type = "b", lty = 2)
> grid()
> lines(variogramLine(vmf, maxdist = 1.3))
> lines(variogramLine(vvmf, maxdist = 1.3), col = "red")
```



Q21 : *Describe the differences between the two empirical semivariograms and also between the two fitted models.* [Jump to A21](#) •

Task 19 : Compute the cross-validation predictions with the two variograms and compare the summary statistics of the residuals. •

We already have the cross-validation object from the original variogram (`k.cv`), computed in §3 above; so we just need to compute a cross-validation object, using `krige.cv` and the second variogram.

```
> kv.cv <- krige.cv(Co ~ 1, loc = jura.cal, model = vvmf)
```

Now compare the cross-validation residuals:

```
> summary(k.cv$residual - kv.cv$residual)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.038500	-0.005000	0.000238	0.001350	0.007850	0.043000

Q22 : Does the use of the second variogram bias the residuals? What is the maximum difference, both absolutely and as a proportion? [Jump to A22](#) •

Task 20 : Clean up from this section. •

```
> rm(ix, sv, svv, v, vv, vvmf, diff.psill, diff.range,  
+     diff.nugget, kv.cv)
```

This should convince you that LOOCV with the variogram estimated from all points is a valid measure of model quality.

3.2 Answers

A10 : There is no obvious trend; however the largest negative and positive prediction errors (over-predictions) seem to be somewhat concentrated towards the SE. Since the two extremes are near each other, adding a trend would probably not improve the cross-validation. [Return to Q10](#) •

A11 : MPE is 0.1 mg kg⁻¹ Co; this is about 0.8% of the mean Co value. There are a few large prediction errors: 9.32 mg kg⁻¹ Co and -5.14 mg kg⁻¹ Co.

The RMSE (precision) is 2.09 mg kg⁻¹ Co, only 38% of the IQR of 5.46 mg kg⁻¹ Co and 58% of the sample set standard deviation. [Return to Q11](#) •

A12 : The MSDR is about 1.11, so the actual data is a bit more variable than what is predicted by kriging. This is not a large difference, so we can say that the model captures the variability fairly well. [Return to Q12](#) •

A13 : The MSDR would be higher, because with a lower nugget the kriging

variance would also be lower. This highlights the importance of a realistic nugget to capture the true small-scale variability. [Return to Q13](#) •

A14 : The bias is the intercept: 0.355; it should be zero. The gain is the slope: 0.954; it should be one; the value is substantially lower, that is, at higher values the predictions tend to under-estimate the actual values, and vice-versa; this typical of a smoothing predictor such as kriging. [Return to Q14](#) •

A15 : The QQ-plot shows strong deviation from normality at both tails: the residuals are more extreme than expected in both cases, especially the negative residuals. This indicates both under- and over-prediction; typical of a smoothing predictor such as kriging.

The residuals vs. fitted plots shows evidence of heteroscedascity: much lower variance at low fitted values.

Thus a linear regression to compare actual vs. predicted values is not appropriate. [Return to Q15](#) •

A16 : The probability that rejecting the null hypothesis of no bias is a Type I error is given by the $\Pr(>|\mathfrak{t}|)$ column of the model summary, for the intercept; it is 0.4. Thus the null hypothesis of no bias should not be rejected; the estimates of 0.355 is likely different from zero only by chance. [Return to Q16](#) •

A17 : The 90% confidence interval includes 1, so we can not reject the null hypothesis of no gain, without running at least a 10% risk of being wrong. But it is quite close; almost all the confidence interval is below 1. [Return to Q17](#) •

A18 : The model explains only 65.8 % of the variation in actual values. This is not very precise. [Return to Q18](#) •

A19 : 60 point-pairs were removed from the original 12632; this is 0.47%. [Return to Q19](#) •

A20 : The changes were minimal: the range was increased by 0.008 km from the original value of 1.524 and the structural sill was increased by 0.22%. The only appreciable difference was in the nugget: because we'd removed the most extreme value, this did have an effect, lowering the nugget by 5.6%, but this was relative to an already quite low value, 1.371. [Return to Q20](#) •

A21 : Both the empirical variograms and the models are almost identical when estimated with and without the extreme point. [Return to Q21](#) •

A22 : Yes, there is a slight bias: the median cross-validation residual is 0.00024 lower when using the variogram estimated without the point with largest value. The largest difference is one residual that is 0.043 lower.

Both of these are very small relative to the cross-validation predictions; the smallest of these is 3.24 mg kg⁻¹ Co; so the largest difference is two orders of magnitude smaller.

[Return to Q22](#) •

4 Kriging prediction variance

There is yet another way to evaluate the quality of a kriging prediction. Recall that kriging also computes a kriging variance, i.e. the variance of the prediction itself. This is a measure of uncertainty, which depends only on the model and the point configuration.

Since we have an independent dataset, we can compare the kriging variance at these points (which were not available during kriging) to the actual RMSE of evaluation at these points.

Task 21 : You should still have object `k.val` in your workspace. If not, repeat the task of Exercise 4 §4.2 “Prediction at known points”: use the fitted variogram model to predict at the 100 evaluation points by Ordinary Kriging (OK). •

Note the use of the `objects` method to return a list of the objects currently defined in the R workspace, the `is.element` set method to determine whether a set has a certain member (here, the name `k.val`), and the conditional `if` operator. So, we only re-compute the kriging if necessary.

```
> if (!is.element("k.val", objects())) {  
+   k.val <- krige(Co ~ 1, loc = jura.cal, newdata = jura.val,  
+               model = vmf)  
+ }
```

Task 22 : Summarize the OK prediction **standard deviations**. •

To be comparable to RMSE the variances must be converted to standard deviations:

```
> summary(sqrt(k.val$var1.var))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.41	2.09	2.20	2.16	2.22	2.81

Q23 : What are the mean and maximum OK prediction standard deviation? •

[Jump to A23](#) •

Q24 : We have evaluated the quality of the OK predictions in three ways:

1. Validation;
2. Cross-validation;
3. Kriging variance (or standard deviation) at evaluation points.

Which of these is (are) **internal** to the calibration data set, i.e. use only the calibration set, and which is (are) **external**, i.e. use new evidence? [Jump to A24](#) •

Q25 : Compare the RMSE (estimates of prediction error) for both evaluation and cross-validation to the mean kriging variance from OK. Do they differ? Why? Which of these three gives the most reliable estimate of the quality of the predictions? [Jump to A25](#) •

4.1 Answers

A23 : The mean is 2.17, the maximum 2.81 mg kg⁻¹ Co. [Return to Q23](#) •

A24 : Validation with a separate evaluation dataset is external, cross-validation is wholly internal, kriging variance at independent points is internal to OK but the selection of points to summarize is external. [Return to Q24](#) •

A25 : The mean OK prediction standard deviation is 2.17 mg kg⁻¹ Co, compared to the evaluation precision of 2.49 mg kg⁻¹ Co and the cross-validation precision of 2.09 mg kg⁻¹ Co. These are of similar magnitude, although the RMSE for evaluation is indeed somewhat higher (perhaps more realistic); both OK and its cross-validation have no independent evaluation. [Return to Q25](#) •

5 Conditional simulation

There are several obvious problems with kriging predictions over an area, with respect to uncertainty:

1. Kriging is an **exact predictor** at known points, because all the weight is given to the known point; this is mathematically necessary but not realistic, since just away from the point the predictions are weighted;
2. Kriging prediction maps are by definition **smooth**, even if there is a nugget component to the model; the actual spatial field is usually much rougher.

Both of these problems are serious issues for **spatially-distributed models**. An example is the soil hydraulic conductivity in models of groundwater flow: the actual erratic nature of this attribute leads to much different model outputs than a smoothly-varying field.

So, a map produced by kriging gives an unrealistic view of the fine-scale spatial variability. We can recover this with **conditional simulation**: this shows one (or many) possible **realizations** of the spatial field:

- as defined by the covariance structure (variogram); and
- as constrained by the known data values.

There are various geostatistical simulation algorithms; that used in `gstat` is described by Pebesma [2].

Many simulated fields can be created, each equally valid, and used as model inputs.

Task 23 : Make four realizations of a conditional simulation of the Co concentration in Jura soils over the prediction grid `jura.grid`, using the OK model. •

The `krige` method can also do conditional simulations. It requires one optional argument:

- `nsim`, the number of conditional simulations

! → Large simulations are quite time-consuming, so the `nmax` and/or `maxdist` optional arguments are often specified as well.

For each set of simulations in this section we use `set.seed` here so your results will look the same as ours; in practice you would only do this if you want to make sure to have the same starting point.

```
> set.seed = 621
> k.sim.4 <- krige(Co ~ 1, loc = jura.cal, newdata = jura.grid,
+   model = vmf, nsim = 4, nmax = 128)
```

Q26 : Compare the means, medians, IQR and ranges of the four simulations, also with the calibration data set on which the simulation was based. *Jump to A26* •

```
> summary(k.sim.4)

Object of class SpatialPointsDataFrame
Coordinates:
      min    max
s1 0.325 5.075
s2 0.425 5.725
Is projected: NA
proj4string : [NA]
Number of points: 10272
Data attributes:
      sim1      sim2      sim3      sim4
Min.   :-2.59   Min.   :-1.84   Min.   :-3.34   Min.   :-2.02
1st Qu.: 6.91   1st Qu.: 7.56   1st Qu.: 7.86   1st Qu.: 7.48
Median : 9.38   Median :10.03   Median :10.29   Median :10.06
Mean   : 9.12   Mean   : 9.99   Mean   :10.24   Mean   : 9.76
3rd Qu.:11.51   3rd Qu.:12.41   3rd Qu.:12.60   3rd Qu.:12.27
Max.   :19.44   Max.   :23.59   Max.   :23.36   Max.   :20.98

> summary(jura.cal$Co)

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.55    6.52    9.76    9.30   12.00   17.70
```


Note: Your results from the four simulations will be different from those shown here. This is because it is a random simulation.

Task 24 : Display the four conditional simulations in one plot, with the locations of the the observation points over-printed. •

The `spplot` method can show several variables at once; in this case we ask for all four fields, which are the four simulations. We also use the optional `sp.layout` argument to specify ‘layout items’ within each plot; in this case the observation points.

For complete grids such as this prediction, `spplot` produces better output when the object to be printed is of class `SpatialPixelsDataFrame`, so we convert by specifying the full grid with `gridded`.¹

```
> class(k.sim.4)

[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"

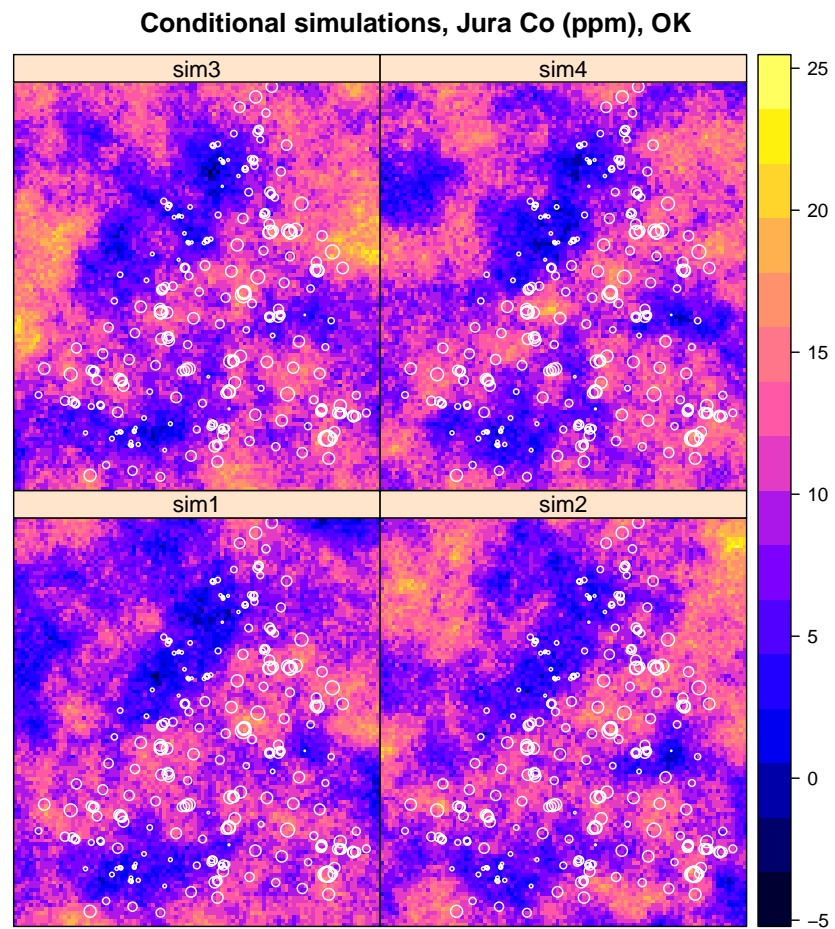
> gridded(k.sim.4) <- TRUE
> class(k.sim.4)

[1] "SpatialPixelsDataFrame"
attr(,"package")
[1] "sp"
```

Note: For correct visualization, when several maps purporting to show the same prediction are shown together, they should have the same scales, in this case a colour ramp. Fortunately, `spplot` does this automatically if several grids are plotted together

```
> layout.2 <- list("sp.points", jura.cal, pch = 1, cex = 1.4 *
+   jura.cal$Co/max(jura.cal$Co), col = "white")
> print(spplot(k.sim.4, zcol = 1:4, col.regions = bpy.colors(64),
+   main = "Conditional simulations, Jura Co (ppm), OK",
+   sp.layout = list(layout.2)))
```

¹ This may not be necessary in updated versions of `gstat`



Q27 : Are the four simulations the same? Do they have similar spatial structure? Are they more similar near the observation points or away from them? What are the differences? Jump to A27 •

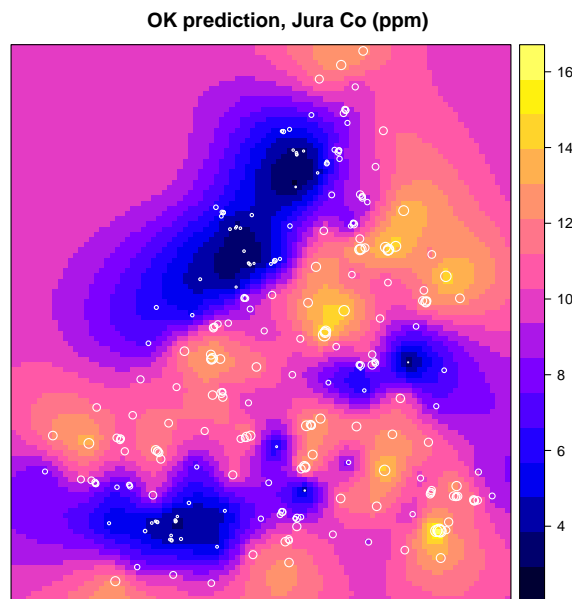
Task 25 : Compare the simulations to the “best single” prediction, i.e., OK. •

To compare the simulations with OK, we re-create the OK prediction and display this best point-wise prediction:

```
> k.ok <- krige(Co ~ 1, loc = jura.cal, newdata = jura.grid,
+   model = vmf)

[using ordinary kriging]

> print(spplot(k.ok, zcol = "var1.pred", col.regions = bpy.colors(64),
+   main = "OK prediction, Jura Co (ppm)", sp.layout = list(layout.2)))
> rm(layout.2)
```



Q28 : *To what extent do the conditional simulations resemble the OK prediction? What is the major difference?* *Jump to A28* •

Simulation is supposed to respect the spatial structure; that is, the simulated field should have the same structure as the variogram model. Let's see how well it was reproduced.

Task 26 : Compute the empirical variograms of the four simulations, and plot them, each with the fitted model `vmf`. •

Note: For correct visualization, when several graphs purporting to show the same thing are shown together, they should have the same scales. So we first compute all four, then find the maximum of all four, as well as the total sill of the variogram model which this is supposed to simulate (using `max`), round this to the next highest integer (using `ceiling`), and use this for the vertical scale; the horizontal is already the same.

We do *not* ask for `plot.numbers=T` because there are so many in this complete field.

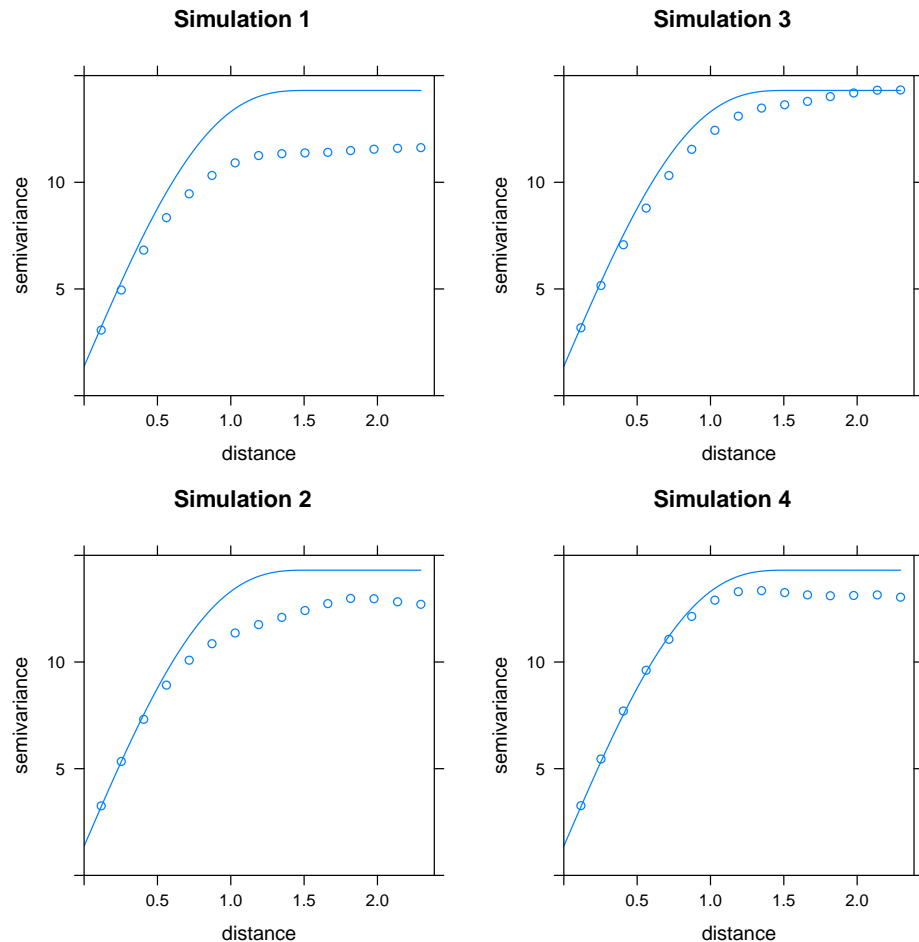
```
> vs.1 <- variogram(sim1 ~ 1, loc = k.sim.4)
> vs.2 <- variogram(sim2 ~ 1, loc = k.sim.4)
> vs.3 <- variogram(sim3 ~ 1, loc = k.sim.4)
> vs.4 <- variogram(sim4 ~ 1, loc = k.sim.4)
> gamma.max = ceiling(max(vs.1$gamma, vs.2$gamma, vs.3$gamma,
+   vs.4$gamma, sum(vmf[, "psill"])))
> p.1 <- plot(vs.1, model = vmf, ylim = c(0, gamma.max),
+   main = "Simulation 1")
> p.2 <- plot(vs.2, model = vmf, ylim = c(0, gamma.max),
+   main = "Simulation 2")
> p.3 <- plot(vs.3, model = vmf, ylim = c(0, gamma.max),
+   main = "Simulation 3")
```

```

> p.4 <- plot(vs.4, model = vmf, ylim = c(0, gamma.max),
+   main = "Simulation 4")

> print(p.1, split = c(1, 1, 2, 2), more = T)
> print(p.2, split = c(1, 2, 2, 2), more = T)
> print(p.3, split = c(2, 1, 2, 2), more = T)
> print(p.4, split = c(2, 2, 2, 2), more = F)

```



```

> rm(vs.1, vs.2, vs.3, vs.4, p.1, p.2, p.3, p.4, gamma.max)

```

Q29 : Do the empirical variograms of the four simulations match the variogram model? What are the differences? Why?

[Jump to A29](#)

•

5.1 Answers

A26 : The values of all of these vary considerably, although they are similar. Note that it is possible to get values outside the original range of samples, in this case even negative (unphysical) values; in this sense kriging is not a **convex** predictor.

[Return to Q26](#) •

A27 : They are not the same, they differ quite a bit from each other. However, near the observation points they are similar; the largest differences are in the corners away from the observations. [Return to Q27](#) •

A28 : The overall pattern of “hot” and “cold” spots is the same within the convex hull of the observation points; for example the “cold” spot in the SW. Away from the points there is no resemblance: the OK prediction is the spatial mean, whereas the simulation reproduces a spatial field with the covariance structure given by the variogram model (see §6 below). The OK prediction is smooth whereas the simulation is noisy, due to the nugget variance and near-linear short-range rise to the (high) partial sill. [Return to Q28](#) •

A29 : It fits almost exactly up to about 0.5 km, and then varies, depending on the exact simulation. This is because of the use of the `nmax` optional argument; approximately 0.5 km is the distance at which this maximum is reached. Recall the grid spacing is 50 m; so that in a square of 0.5 km² there are 10 x 10 = 100 cells. [Return to Q29](#) •

6 * Unconditional simulation

In some situations we would like to simulate a random field defined by an assumed spatial covariance structure (e.g., as represented by a variogram model), without considering any data points. For example, for designing a sampling scheme, if we have an idea of the field structure, we can create many realizations by **unconditional simulation**, and see how well our proposed scheme would capture the known structure.

Task 27 : Make four realizations of a unconditional simulation of the Co concentration in Jura soils over the prediction grid `jura.grid`, using the OK model. •

This is as the previous §5, using the `krige` method. However, for unconditional simulation we specify:

1. that the data locations are missing, by setting the `loc` argument to `NULL`;
2. that there is no data, by setting the `dummy` argument to `TRUE`;
3. the value of the trend surface coefficients, by setting the `beta` argument to the presumed mean; for OK this is the mean, so that OK becomes Simple Kriging (SK).

As with conditional simulation, the `nmax` and/or `maxdist` optional arguments are often specified as well.

The `beta` argument gives the expected value of the stationary field; here it is most reasonably set to the non-spatial mean of the Co concentration at

the sample points.

Note: More properly, the spatial mean could be used.

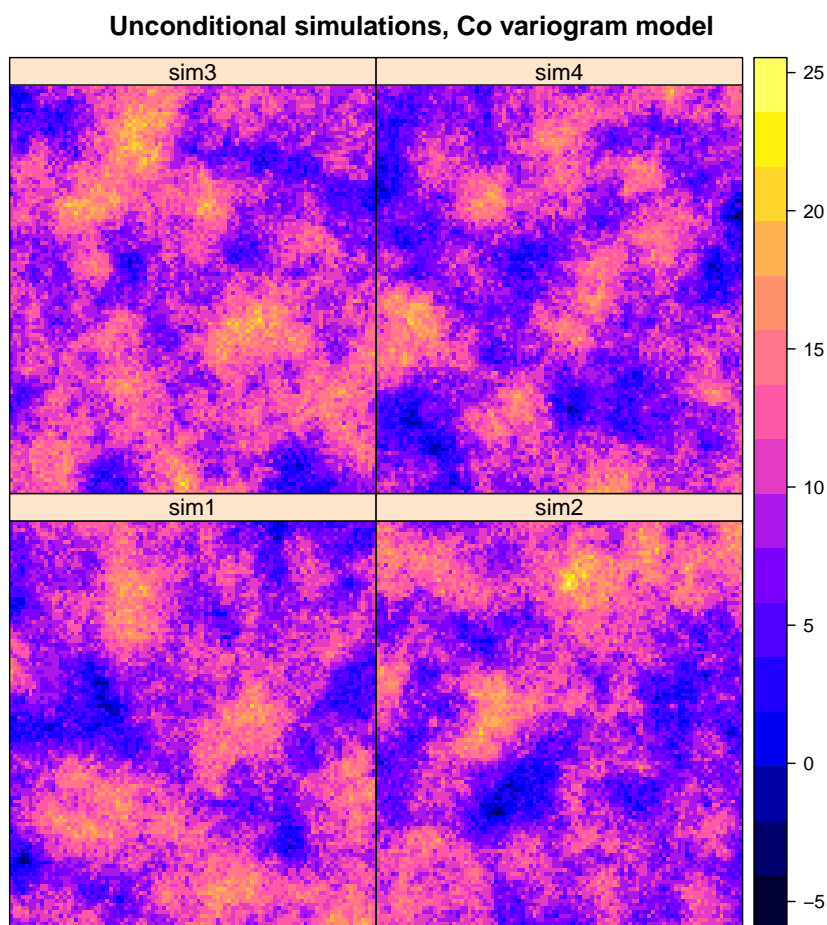
```
> set.seed(621)
> k.sim.4.u <- krige(z ~ 1, loc = NULL, newdata = jura.grid,
+   model = vmf, nsim = 4, nmax = 128, beta = mean(jura.cal$Co),
+   dummy = T)
```

Task 28 : Compare the means, medians, IQR and ranges of the four simulations. •

```
> summary(k.sim.4.u)

Object of class SpatialGridDataFrame
Coordinates:
      min max
[1,] 0.3 5.10
[2,] 0.4 5.75
Is projected: NA
proj4string : [NA]
Grid attributes:
  cellcentre.offset cellsize cells.dim
1           0.325      0.05         96
2           0.425      0.05        107
Data attributes:
      sim1      sim2      sim3      sim4
Min.   :-3.94  Min.   :-3.77  Min.   :-1.28  Min.   :-4.07
1st Qu.: 7.27  1st Qu.: 7.11  1st Qu.: 8.12  1st Qu.: 6.61
Median : 9.68  Median : 9.53  Median :10.38  Median : 8.92
Mean   : 9.61  Mean   : 9.53  Mean   :10.34  Mean   : 8.94
3rd Qu.:12.01  3rd Qu.:11.90  3rd Qu.:12.60  3rd Qu.:11.27
Max.   :20.39  Max.   :23.61  Max.   :22.72  Max.   :20.76

> print(spplot(k.sim.4.u, zcol = 1:4, col.regions = bpy.colors(64),
+   main = "Unconditional simulations, Co variogram model"))
```



Q30 : *Describe the similarities and differences among the four simulations.*
[Jump to A30](#) •

Q31 : *How do these differ from the conditional simulations with the same variogram model and same interpolation grid?*
[Jump to A31](#) •

One use of unconditional simulation is to **visualize different random fields**.

We begin by comparing random fields with the same variogram parameters but different forms.

Task 29 : Simulate random fields on the Jura grid for spherical, pentaspherical, exponential, Gaussian models, all with the same effective range, partial sill and nugget as the fitted model for Jura cobalt. •

Recall that the effective range of the exponential model is 3x the range parameter, and for the Gaussian model it is $\sqrt{3}$ times.

```
> c0 <- vmf[1,"psill"]; c1 <- vmf[2,"psill"]; a <- vmf[2,"range"]
```

```

> set.seed(621)
> k.sim.u.sph <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                     model=vgm(c1,"Sph",a,c0), nsim=1, nmax=128,
+                     beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.pen <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                     model=vgm(c1,"Pen",a,c0), nsim=1, nmax=128,
+                     beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.exp <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                     model=vgm(c1,"Exp",a/3,c0), nsim=1, nmax=128,
+                     beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.gau <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                     model=vgm(c1,"Gau",a/sqrt(3),c0), nsim=1,
+                     nmax=128, beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

```

Task 30 : Display the simulated fields in one figure; also display the variograms that produced them. •

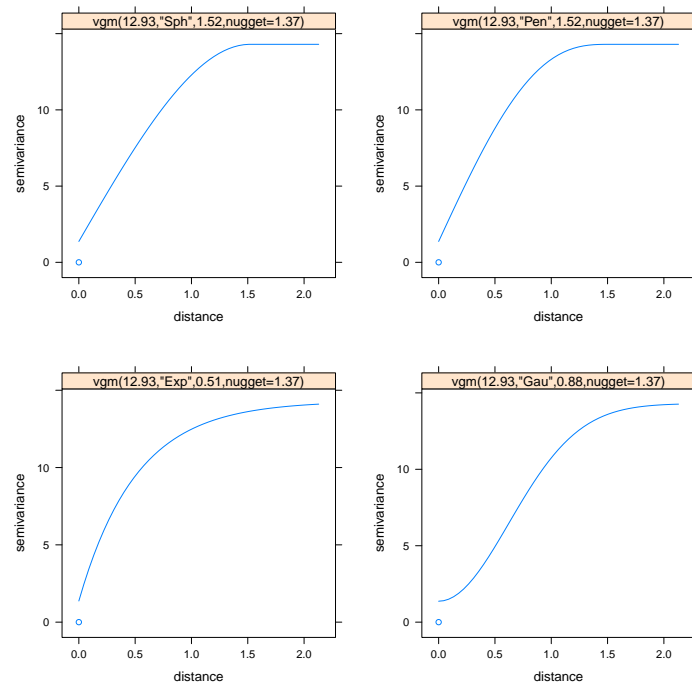
The `range` parameter for the Gaussian and exponential models must be adjusted in the same way as for the simulations just above.

```

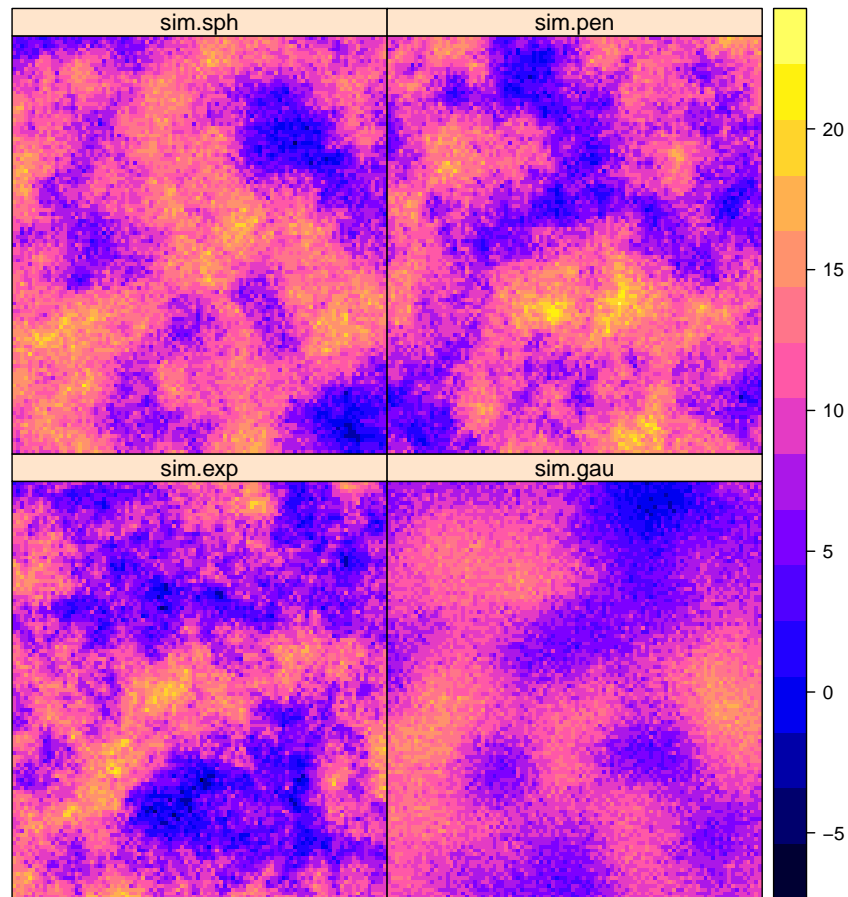
> p.1 <- show.vgms(max = round(a * 1.4, 2), sill = round(c1,
+ 2), range = round(a, 2), nugget = round(c0, 2), models = "Sph")
> p.2 <- show.vgms(max = round(a * 1.4, 2), sill = round(c1,
+ 2), range = round(a, 2), nugget = round(c0, 2), models = "Pen")
> p.3 <- show.vgms(max = round(a * 1.4, 2), sill = round(c1,
+ 2), range = round(a/3, 2), nugget = round(c0, 2),
+ models = "Exp")
> p.4 <- show.vgms(max = round(a * 1.4, 2), sill = round(c1,
+ 2), range = round(a/sqrt(3), 2), nugget = round(c0,
+ 2), models = "Gau")

> print(p.1, split = c(1, 1, 2, 2), more = T)
> print(p.2, split = c(2, 1, 2, 2), more = T)
> print(p.3, split = c(1, 2, 2, 2), more = T)
> print(p.4, split = c(2, 2, 2, 2), more = F)

```

```
> k.sim.u <- k.sim.u.sph
> names(k.sim.u) <- "sim.sph"
> k.sim.u$sim.pen <- k.sim.u.pen$sim1
> k.sim.u$sim.exp <- k.sim.u.exp$sim1
> k.sim.u$sim.gau <- k.sim.u.gau$sim1
> spplot(k.sim.u, zcol = c(3, 4, 1, 2), col.regions = bpy.colors(64))
```



Q32 : Describe the differences between the fields simulated from the four models. *Jump to A32 •*

Another interesting visualization is the same model but with increasing proportion of nugget variance.

Task 31 : Simulate random fields on the Jura grid for a spherical model with a single range and total sill as the fitted model for Jura cobalt, but with four different nugget-to-total sill ratios: 0, 0.2, 0.4, 0.6. •

We use the same variogram model, specifying different proportions of the total sill for the structural (partial) sill and the nugget:

```
> c.total <- sum(vmf[, "psill"]); a <- vmf[2, "range"]
> set.seed(621)
> k.sim.u.sph.0 <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                       model=vgm(c.total, "Sph", a, 0), nsim=1,
+                       nmax=128, beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.sph.2 <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
```

```

+                               model=vgm(c.total*.8,"Sph",a,c.total*.2), nsim=1,
+                               nmax=128, beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.sph.4 <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                       model=vgm(c.total*.6,"Sph",a,c.total*.4), nsim=1,
+                       nmax=128, beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

> k.sim.u.sph.6 <- krige(z ~ 1, loc=NULL, newdata=jura.grid,
+                       model=vgm(c.total*.4,"Sph",a,c.total*.6), nsim=1,
+                       nmax=128, beta=mean(jura.cal$Co), dummy=T)

[using unconditional Gaussian simulation]

```

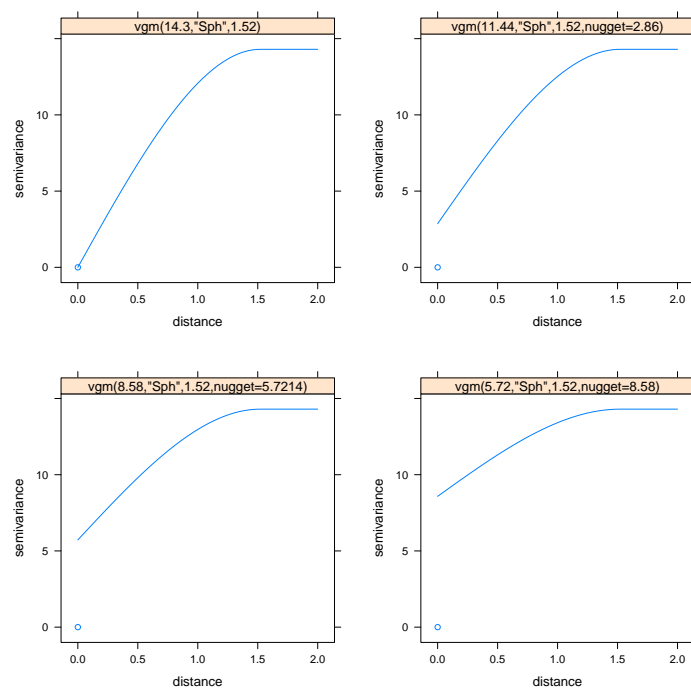
Task 32 : Display the simulated fields in one figure; also display the variograms that produced them. •

```

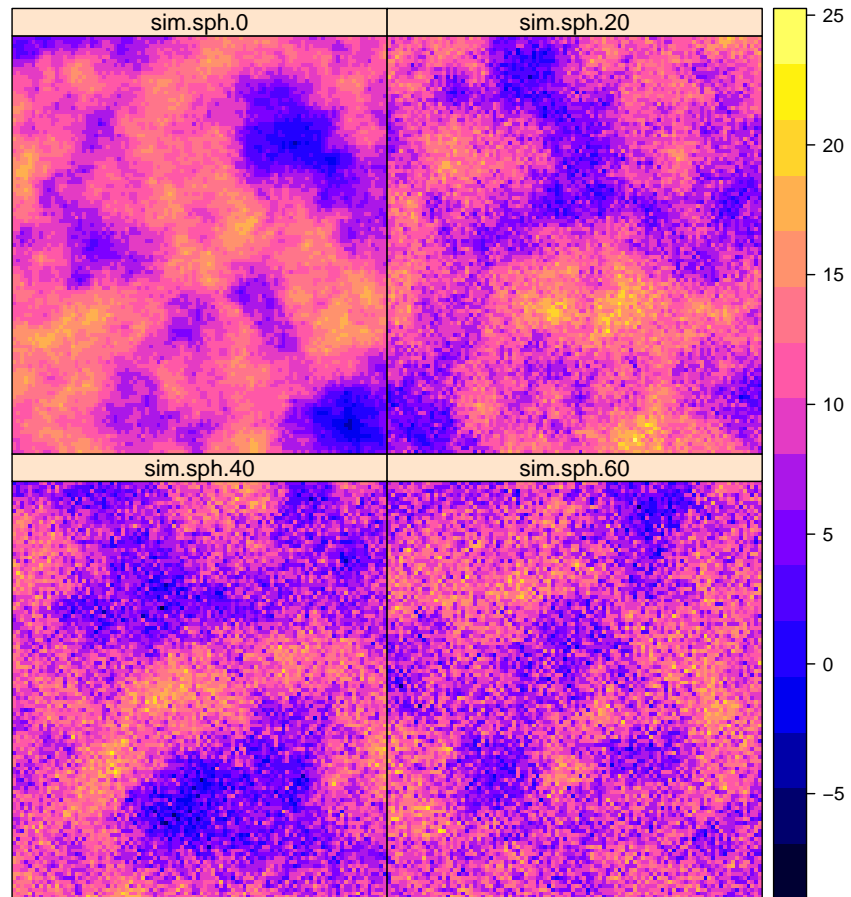
> p.1 <- show.vgms(max = round(a * 1.2), sill = round(c.total,
+ 2), range = round(a, 2), models = "Sph")
> p.2 <- show.vgms(max = round(a * 1.2), sill = round(c.total *
+ 0.8, 2), nugget = round(c.total * 0.2, 2), range = round(a,
+ 2), models = "Sph")
> p.3 <- show.vgms(max = round(a * 1.2), sill = round(c.total *
+ 0.6, 2), nugget = round(c.total * 0.4, 4), range = round(a,
+ 2), models = "Sph")
> p.4 <- show.vgms(max = round(a * 1.2), sill = round(c.total *
+ 0.4, 2), nugget = round(c.total * 0.6, 2), range = round(a,
+ 2), models = "Sph")

> print(p.1, split = c(1, 1, 2, 2), more = T)
> print(p.2, split = c(2, 1, 2, 2), more = T)
> print(p.3, split = c(1, 2, 2, 2), more = T)
> print(p.4, split = c(2, 2, 2, 2), more = F)

```



```
> k.sim.u <- k.sim.u.sph.0
> names(k.sim.u) <- "sim.sph.0"
> k.sim.u$sim.sph.20 <- k.sim.u.sph.2$sim1
> k.sim.u$sim.sph.40 <- k.sim.u.sph.4$sim1
> k.sim.u$sim.sph.60 <- k.sim.u.sph.6$sim1
> spplot(k.sim.u, zcol = c(3, 4, 1, 2), col.regions = bpy.colors(64))
```



Q33 : Describe the differences between the fields simulated the four model with the same range and total sill, but different proportions of nugget variances. *Jump to A33*

•

Task 33 : Clean up from this section. •

```
> rm(k.sim.4.u, k.sim.u, k.sim.u.sph, k.sim.u.pen, k.sim.u.exp,
+    k.sim.u.gau, k.sim.u.sph.0, k.sim.u.sph.2, k.sim.u.sph.4,
+    k.sim.u.sph.6, c0, c1, c.total, a)
> rm(p.1, p.2, p.3, p.4)
```

6.1 Answers

A30 : The four simulations have the same spatial structure, i.e., size, shape and continuity of patches; however they differ in where different patches are located, and in details within each patch. *Return to Q30* •

A31 : The unconditional simulations have no relation to the data values at the known points. High or low values can occur anywhere. [Return to Q31](#) •

A32 : The Gaussian model has the largest patches, with strong spatial continuity. The exponential model shows the least continuity, with no clear edges to patches. The spherical and pentaspherical models are quite similar; the latter should have slightly wider transition zones between patches, due to the more gradual “shoulder” in the variogram model. [Return to Q32](#) •

A33 : Increasing the nugget variance increases the noise in the simulated field. When the nugget is 60% of the total sill the pattern is strongly masked by the noise.. When there is no nugget, the patches are quite clear. The others are intermediate. [Return to Q33](#) •

7 Self-test

This section is a small self-test of how well you mastered this exercise. You should be able to complete the tasks and answer the questions with the knowledge you have gained from the exercise. Please **submit your answers (including graphical output) to the instructor** for grading and sample answers.

Task 1 : Repeat the conditional simulation of the Jura cobalt concentration, but with the 100 evaluation points in place of the 259 calibration points as observations. •

Q1 : *Compare this simulation with the simulation using the calibration points. Which one is more spatially variable among the four realizations? In other words, in which simulation set do the four realizations look more like each other? Why is this?* •

Task 2 : If you completed optional §6, simulate a random field with one model but either: (1) four different ranges; (2) four different partial (structural) sills; or (3) four different nugget variances. Summarize the statistics and display the simulated fields. •

Q2 : *Describe the differences that varying your chosen parameter makes on the simulated field. Describe how that fits with the theory of random fields.* •

References

- [1] M. Analla. Model validation through the linear regression fit to actual versus predicted values. *Agricultural Systems*, 57(1):115–119, 1998. 4
- [2] E J Pebesma. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7):683–691, 2004. 22

Index of R Concepts

`as.data.frame`, 2
`asp=1` graphics argument, 4

`beta` gstat argument, 27
`bubble` (`package:sp`), 2

`ceiling`, 4, 25
`coef`, 7
`coordinates` (`package:sp`), 2

`dummy` gstat argument, 27

`floor`, 4

`grid`, 17
`gridded` (`package:sp`), 23
`gstat` package, 10, 17, 22, 23

`hist`, 2

`if`, 20
`IQR`, 3
`is.element`, 20

`krige` (`package:gstat`), 22, 27
`krige.cv` (`package:gstat`), 10, 11, 18

`lines`, 17
`lm` class, 7
`loc` gstat argument, 27

`max`, 4, 25
`min`, 4

`objects`, 20

`plot`, 17
`plot.xy`, 4
`points`, 17

`qt`, 8

`set.seed`, 22
`SpatialPixelsDataFrame` class, 23
`spplot` (`package:sp`), 23
`summary`, 2, 7
`summary.lm`, 7

`title`, 17

`variogramLine` (`package:gstat`), 17