
Applied geostatistics

Exercise 4a: Normal-score transformation

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

January 3, 2014

Contents

1	Introduction	1
2	Normal-score transformation	1
3	Variogram modelling	5
4	Simple Kriging	7
5	Back-transformation	9
6	Comparison with Ordinary Kriging	12
7	Answers	13
	References	15
	Index of R concepts	17

1 Introduction

Goovaerts [1, §7.2.2] explains the rationale behind the **normal score transform**. Here we present some of that rationale and show how to implement the transform in R with the `gstat` package.

Task 1 : If R is not already running, start it. If you haven't already done so, load the `gstat` and `sp` libraries, as shown in the previous exercises. •

```
> require(sp)
> require(gstat)
> require(lattice)
```

Task 2 : If the `jura.cal` spatial object is not already in the workspace, load it from the saved image. •

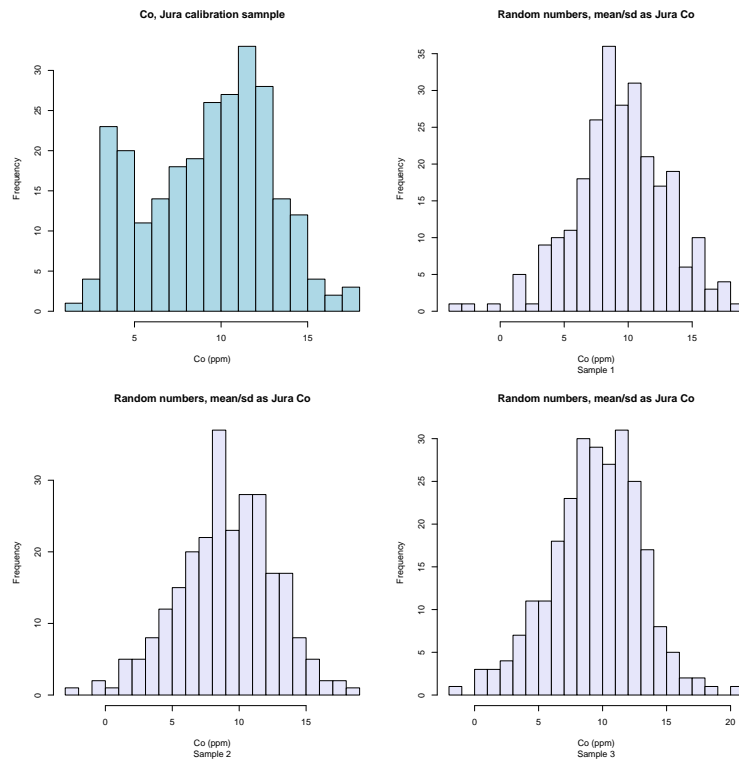
```
> load("JuraEx4.RData")
```

2 Normal-score transformation

Multigaussian kriging requires that the variable to be modelled and kriged be normally-distributed.

Task 3 : Display a histogram of the Co values, and three sample histogram of the same number of random normal values with the empirical mean and standard deviation. •

```
> par(mfrow = c(2, 2))
> hist(jura.cal$Co, col = "lightblue", breaks = 20,
+      main = "Co, Jura calibration sample", xlab = "Co (ppm)")
> for (i in 1:3) hist(rnorm(length(jura.cal$Co), mean = mean(jura.cal$Co),
+      sd = sd(jura.cal$Co)), col = "lavender", breaks = 20,
+      main = "Random numbers, mean/sd as Jura Co",
+      xlab = "Co (ppm)", sub = paste("Sample", i))
> par(mfrow = c(1, 1))
```



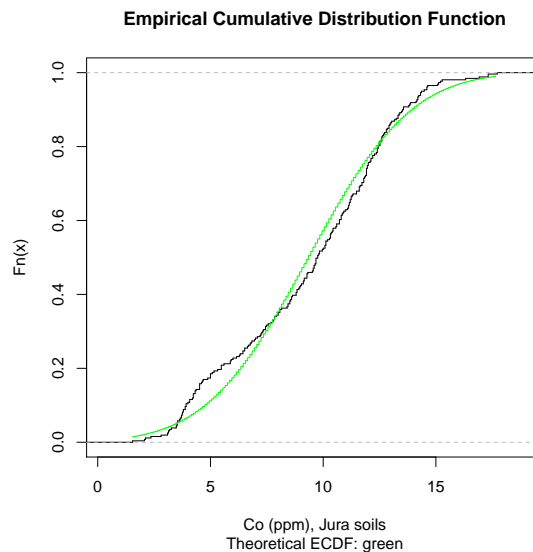
Q1 : *Is the empirical distribution approximately normal?* [Jump to A1](#) •

We can see this also in a cumulative probability plot.

Task 4 : Plot the Empirical Cumulative Distribution Function (ECDF) of the Jura calibration sample Co concentrations. Superimpose the theoretical ECDF for a normal distribution with the same mean and standard deviation.

For the empirical plot, we use the `ecdf` “Empirical Cumulative Distribution Function” method. For the theoretical plot, we use the `seq` method to create a vector of Co concentrations, and then the `pnorm` method to compute the cumulative probability of achieving each value. Plotting the theoretical curve with the `points` method adds to the existing plot; specifying `type="s"` gives a **step function**.

```
> plot(ecdf(jura.cal$Co), do.points = F, verticals = T,
+      main = "Empirical Cumulative Distribution Function",
+      xlab = "Co (ppm), Jura soils", sub = "Theoretical ECDF: green")
> xvals <- seq(min(jura.cal$Co), max(jura.cal$Co),
+      by = 0.1)
> points(x = xvals, y = pnorm(xvals, mean = mean(jura.cal$Co),
+      sd = sd(jura.cal$Co)), col = "green", type = "s")
```



Q2 : *What are the principal differences between the actual and theoretical (normal) ECDF?* Jump to A2 •

Q3 : *Would a monotonic function (such as logarithm or square root) transform this empirical distribution to approximate normality?* Jump to A3 •

Since no monotonic function will transform these, we use instead the **normal scores**, i.e. the quantile of the normal distribution with the observed mean and standard deviation.

Task 5 : Compute the normal scores for the Co values. •

We can see the correspondence between the original values and their normal scores with the `qqnorm` method, which by default plots a normal QQ plot, but also can return two vectors: the original values and their normal scores.

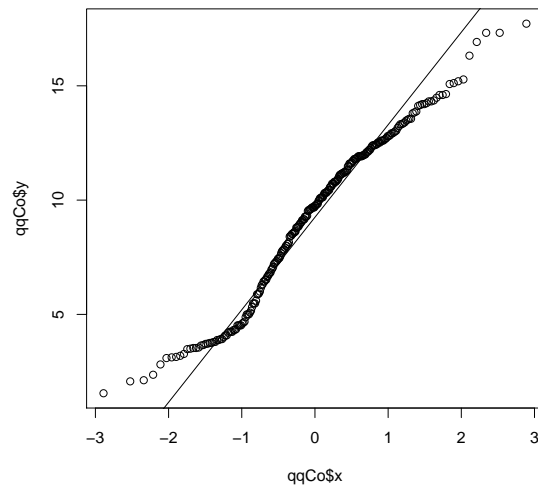
```
> qqCo <- qqnorm(jura.cal$Co, plot.it = F)
> str(qqCo)

List of 2
 $ x: num [1:259] -0.1067 0.0581 0.2146 0.5946 2.1113 ...
 $ y: num [1:259] 9.32 10 10.6 11.92 16.32 ...
```

The normal scores are in the `x` field, the original Co values in the `y` field.

We can compare this to a theoretical normal distribution by plotting the results of the `qqnorm` method; this is equivalent to calling `qqnorm(jura.cal$Co)` directly. The theoretical line is superimposed with the `qqline` method.

```
> plot(qqCo)
> qqline(jura.cal$Co)
```



Examine the correspondence between value and normal score numerically, at both extremes:

```
> head(sort(qqCo$x))

[1] -2.8893 -2.5246 -2.3396 -2.2111 -2.1113 -2.0289

> head(sort(qqCo$y))

[1] 1.55 2.07 2.12 2.36 2.81 3.09

> tail(sort(qqCo$x))

[1] 2.0289 2.1113 2.2111 2.3396 2.5246 2.8893

> tail(sort(qqCo$y))

[1] 15.28 16.32 16.92 17.32 17.32 17.72
```

Task 6 : Make a data frame with the correspondence between Co concentration and normal score, sorted from lowest to highest quantile. •

We use the `order` method to rank the scores (or, equivalently, the concentrations), and then use this index to place the scores and concentrations in sorted order in the dataframe:

```
> head(order(qqCo$x))

[1] 84 114 48 13 129 123

> head(order(qqCo$y))

[1] 84 114 48 13 129 123

> qqCo.s <- data.frame(score = qqCo$x[order(qqCo$y)],
+   Co = qqCo$y[order(qqCo$y)])
> str(qqCo.s)
```

```
'data.frame':      259 obs. of  2 variables:
 $ score: num  -2.89 -2.52 -2.34 -2.21 -2.11 ...
 $ Co   : num   1.55 2.07 2.12 2.36 2.81 3.09 3.12 3.14 3.19 3.27 ...
```

Task 7 : Add a field with the normal score transforms of Co to the calibration spatial object. •

```
> jura.cal$Cd.norm <- qqnorm(jura.cal$Cd, plot.it = F)$x
```

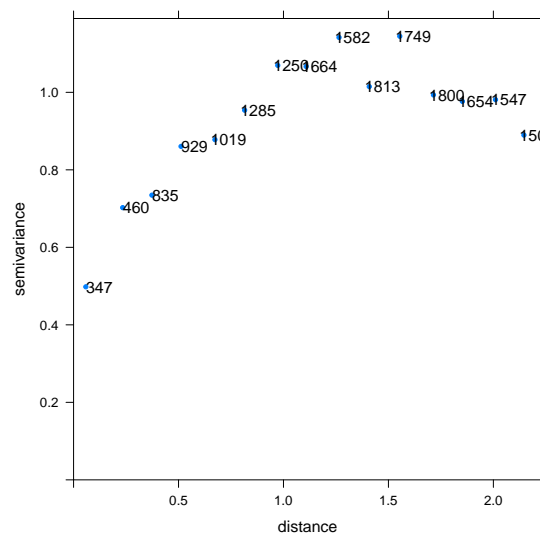
3 Variogram modelling

Now we have a ‘new’ variable and can model its spatial structure.

Task 8 : Compute and model the variogram for the normal score transforms of Co. •

We first compute and plot the variogram:

```
> v <- variogram(Cd.norm ~ 1, jura.cal)
> print(plot(v, pl = T, pch = 20))
```

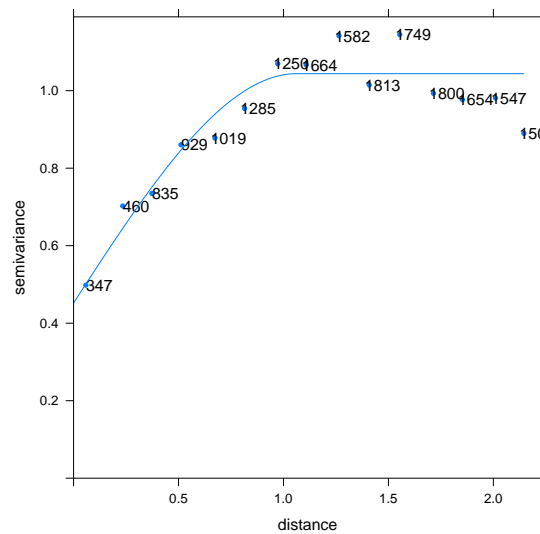


Then, we estimate a model and fit it automatically:

```
> (vmf <- fit.variogram(v, vgm(0.6, "Sph", 1, 0.4)))

model  psill  range
1  Nug 0.45230 0.0000
2  Sph 0.59161 1.0628

> print(plot(v, model = vmf, pl = T, pch = 20))
```



The proportion of variance explained is the structural sill divided by the total sill:

```
> vmf[2, "psill"]/sum(vmf[, "psill"])

[1] 0.56672
```

Task 9: Model the variogram and compute proportion of variance explained for the untransformed variable. •

```
> v.c <- variogram(Co ~ 1, jura.cal)
> (vmf.c <- fit.variogram(v.c, vgm(10, "Sph", 1.2,
+ 5)))

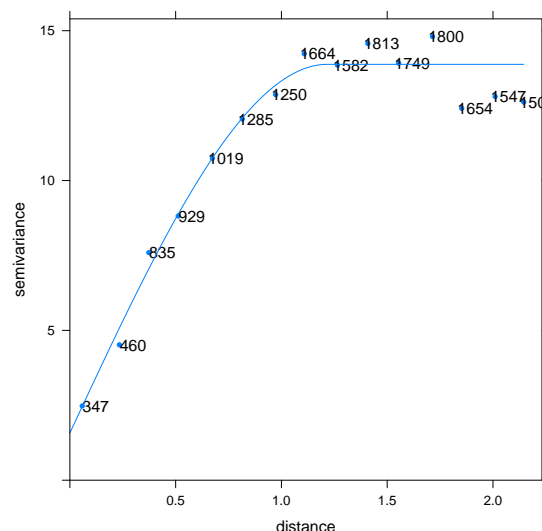
model  psill  range
1  Nug  1.5749 0.0000
2  Sph 12.3033 1.2175

> print(plot(v.c, pl = T, pch = 20, model = vmf.c))
> vmf.c[2, "range"]

[1] 1.2175

> vmf.c[2, "psill"]/sum(vmf.c[, "psill"])

[1] 0.88652
```



Q4 : What is the fitted variogram model? How does the range and proportion of variance explained compare to the variogram model of the untransformed variable? Jump to A4

•

Q5 : Why is the proportion of variance explained by the variogram model so much lower for the normal-score transformed variable? Jump to A5 •

4 Simple Kriging

With a model, we can now predict.

Task 10 : Predict over the grid with the normal-score transform variable, using Simple Kriging (SK). •

We can use SK because the mean is by definition 0 after normal-score transformation.

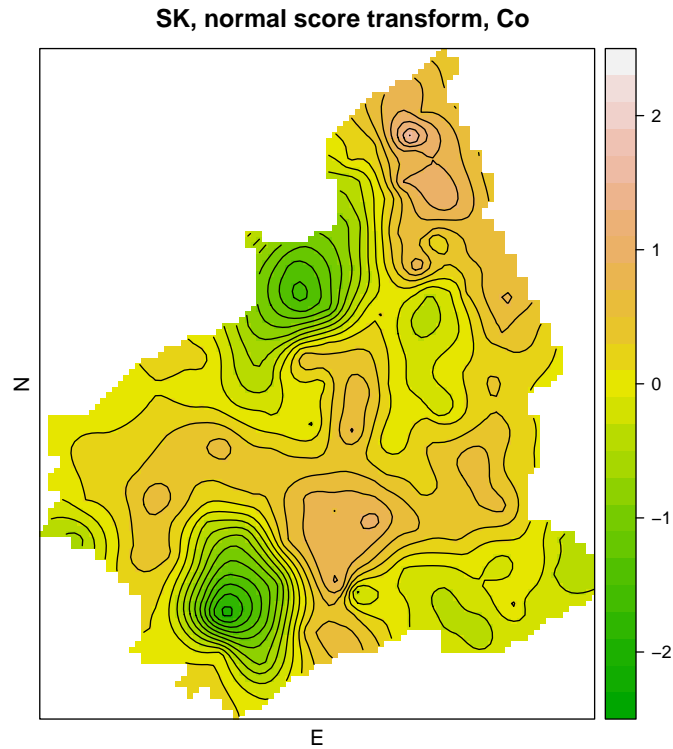
```
> k <- krige(Cd.norm ~ 1, jura.cal, newdata = jura.grid,
+           model = vmf, beta = 0)
```

```
[using simple kriging]
```

```
> summary(k@data)
```

var1.pred	var1.var
Min. :-1.9332	Min. :0.534
1st Qu.: -0.1512	1st Qu.: 0.645
Median : 0.1317	Median : 0.680
Mean : 0.0712	Mean : 0.692
3rd Qu.: 0.3998	3rd Qu.: 0.709
Max. : 1.5042	Max. : 1.039

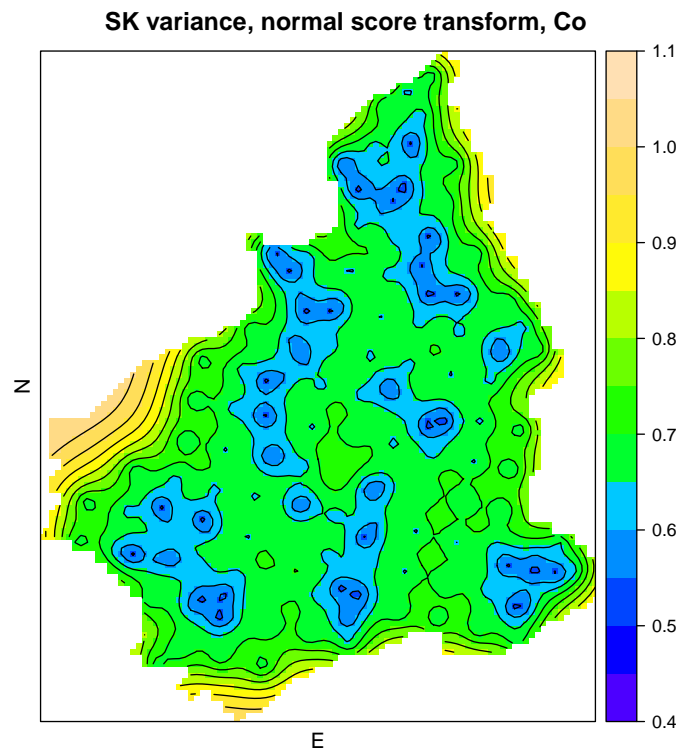

```
> print(spplot(k, z = "var1.pred", contour = T, at = seq(-2.5,
+   +2.5, by = 0.2), col.regions = terrain.colors(64),
+   main = "SK, normal score transform, Co", xlab = "E",
+   ylab = "N"))
```



Q6 : What does the value 0 (zero) on this map represent, in terms of Co concentration? *Jump to A6 •*

We can also plot the kriging prediction variances:

```
> print(spplot(k, z="var1.var", col.regions=topo.colors(64),
+   contour=T,
+   at=seq(0.4,1.1, by=0.05),
+   main="SK variance, normal score transform, Co",
+   xlab="E", ylab="N"))
```



5 Back-transformation

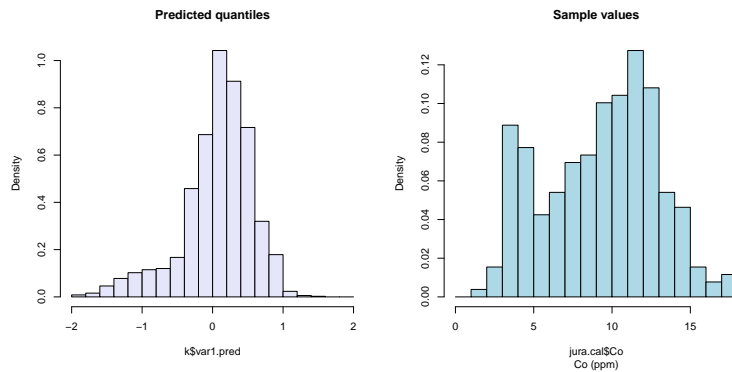
Of course, we want our predictions in terms of the actual element concentration, not a normal score.

Task 11 : Back-transform the predictions to Co values; compare their summary statistics and histogram to the sample's. •

The result of the normal-score SK is a map of normal scores, where 0 represents the mean and other values represent the number of standard deviations above or below that. If the original sample were normally-distributed, we could recover the Co values in original units, we just multiply the deviations by the sample standard deviation (thereby recovering the original units of measure) and add these to the sample mean (thereby recovering the original centre of the distribution location).

However, the whole point of this exercise was that the Co values in the original sample was *not* normally-distributed, rather its histogram was irregular, with several peaks. The discrepancy can be appreciated in the following matched histograms:

```
> par(mfrow = c(1, 2))
> hist(k$var1.pred, main = "Predicted quantiles", breaks = seq(-2,
+   +2, by = 0.2), col = "lavender", freq = F)
> hist(jura.cal$Co, main = "Sample values", sub = "Co (ppm)",
+   breaks = 0:18, col = "lightblue", freq = F)
> par(mfrow = c(1, 1))
```



To solve this problem, we use the correspondence between quantiles and Co concentrations established above, and saved in data frame `qqCo.s`, to perform a **linear interpolation** of Co values, based on the predicted normal scores, using the `approx` method. This takes a two-dimensional vector (here, Co concentration vs. score) and, given a score, estimates the Co concentration by interpolating between the nearest two scores in the correspondence vector. It returns a two-vector list: the abscissa as `x` and the ordinate as `y`.

Here is the structure of the approximation object:

```
> str(approx(qqCo.s$score, qqCo.s$Co, xout = k$var1.pred))

List of 2
 $ x: num [1:5957] -0.343 -0.384 -0.35 -0.423 -0.383 ...
 $ y: num [1:5957] 8.41 8.02 8.3 7.84 8.03 ...
```

Q7 : What is the Co concentration corresponding to a normal score of -1.57? Jump to A7 •

```
> approx(qqCo.s$score, qqCo.s$Co, xout = -1.57)

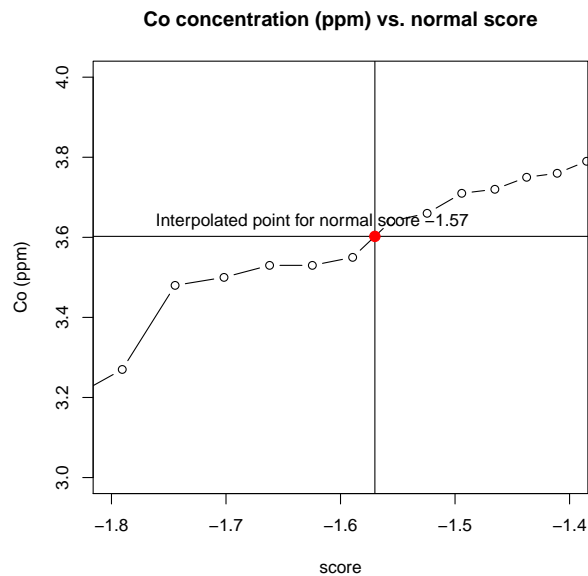
$x
[1] -1.57

$y
[1] 3.6024
```

We can visualize this by seeing one interpolation towards the low tail of the -Q plot. We draw the sample point concentrations and scores, and show the linear interpolation between them:

```
> plot(qqCo.s$Co ~ qqCo.s$score,
+      main="Co concentration (ppm) vs. normal score",
+      xlim=c(-1.8,-1.4), ylim=c(3,4), type="b",
+      xlab="score", ylab="Co (ppm)")
> abline(v=-1.57)
> abline(h=approx(qqCo.s$score, qqCo.s$Co, xout=-1.57))
> points(approx(qqCo.s$score, qqCo.s$Co, xout=-1.57),
+        cex=2, col="red", pch=20)
> text(approx(qqCo.s$score, qqCo.s$Co, xout=-1.57),
```

```
+ "Interpolated point for normal score -1.57",
+ adj=(c(0.7, -0.8)))
```



Now that we have seen how the back-transformation can be implemented, we interpolate all of the predictions and store them in the prediction frame:

```
> k$Co.pred <- approx(qqCo.s$score, qqCo.s$Co, xout = k$var1.pred)$y
> summary(k$Co.pred)
```

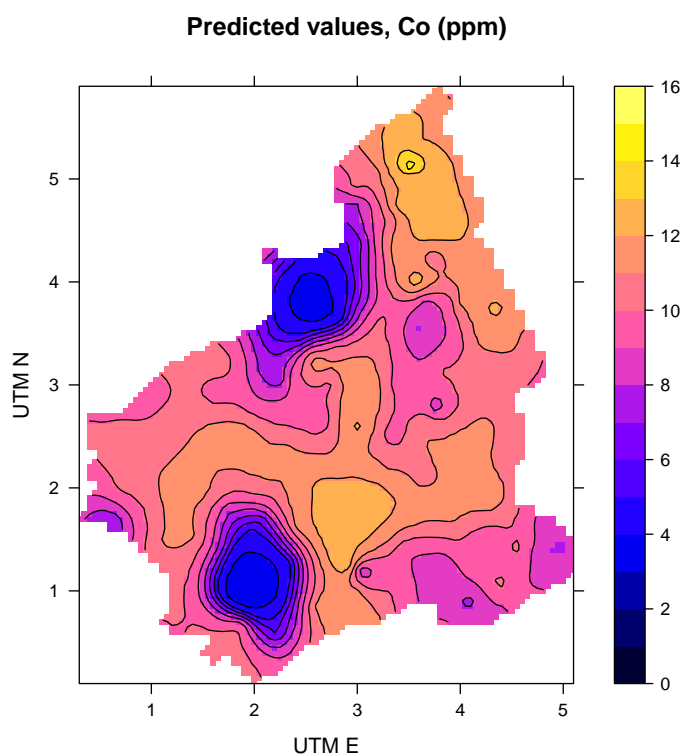
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.13	9.16	10.30	9.87	11.20	14.20

```
> summary(jura.cal$Co)
```

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

Task 12 : Plot the predictions of Co concentration. •

```
> print(spplot(k, zcol="Co.pred", pretty=T, contour=T,
+             at=0:16, col.regions=bpy.colors(64),
+             main="Predicted values, Co (ppm)",
+             xlab="UTM E", ylab="UTM N",
+             scales=list(draw=T)))
```



6 Comparison with Ordinary Kriging

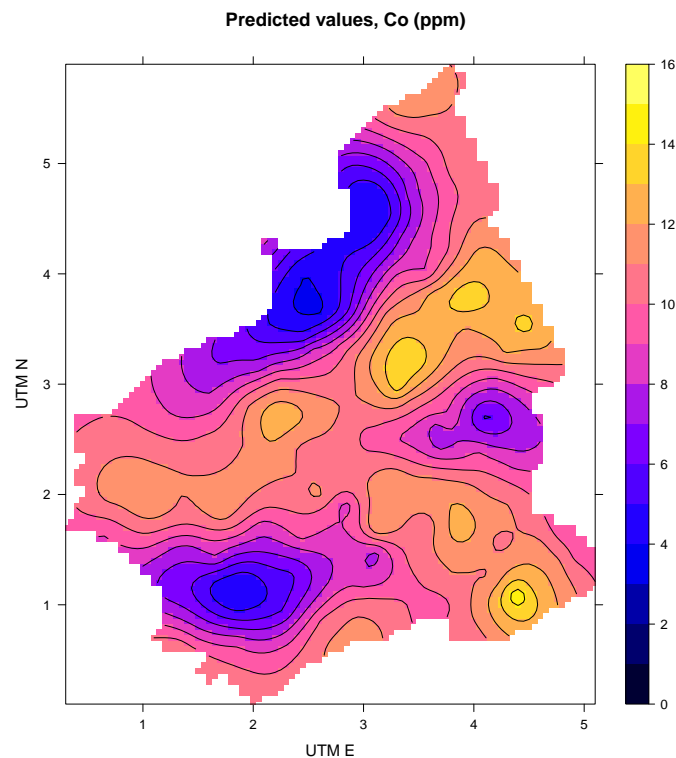
Task 13 : Compare with the OK predictions of the untransformed variable.

We have the fitted variogram model from the comparison above.

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

[using ordinary kriging]

```
> print(spplot(k.grid, zcol = "var1.pred", pretty = T,
+   contour = T, at = 0:16, col.regions = bpy.colors(64),
+   main = "Predicted values, Co (ppm)", xlab = "UTM E",
+   ylab = "UTM N", scales = list(draw = T)))
```



Q8 : What are the major differences between the two predictions? [Jump to A8](#) •

```
> summary(k.grid$var1.pred)

    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.777   8.28   10.10   9.60  11.10  14.60

> summary(k$Co.pred)

    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.13   9.16   10.30   9.87  11.20  14.20
```

Task 14 : Clean up from this exercise. •

```
> rm(xvals, v, vm, vmf, v.c, vmf.c, qqCo, qqCo.s, k,
+    k.grid, i)
```

7 Answers

A1 : No; although the tails are shaped approximately as in a normal distribution, there two clear modes (near 3 and 12 mg kg⁻¹ Co). The three samples from the normal distribution vary considerably among themselves; this illustrates the effect of small (259, in this case) sample size. [Return to Q1](#) •

A2 : The theoretical ECDF has substantially lower probabilities for Co values from about 3 to about 8; this corresponds to the first peak in the empirical histogram.

[Return to Q2](#) •

A3 : No, because the empirical distribution is not monotonically increasing in frequency (i.e. it is multi-modal).

[Return to Q3](#) •

A4 : For the normal-score variogram model: Range 1.06; structural sill 0.592; nugget variance 0.452; proportion of variance explained 0.57.

This range is somewhat shorter than the model for the untransformed variable: 1.22; the proportion of variance explained is much lower than the untransformed variable's 0.89

[Return to Q4](#) •

A5 : The normal scores are on a much narrower range of values: -2.89 to 2.89 for the normal score; 1.55 to 17.72 for the original variable. So the sill must be lower. The higher relative nugget is due to the stretching of the middle part of the range.

[Return to Q5](#) •

A6 : A normal score of 0 (zero) represents the mean of the original variable; in this case 9.3.

[Return to Q6](#) •

A7 : The Co concentration corresponding to a normal score of -1.57 is 3.602.

[Return to Q7](#) •

A8 : The range of predictions is similar but the spatial pattern is quite different. The centres of the “cold spots” are similar but the “hot spots” are not. [Return to Q8](#) •

References

- [1] P. Goovaerts. *Geostatistics for natural resources evaluation*. Applied Geostatistics. Oxford University Press, New York; Oxford, 1997. [1](#)

Index of R Concepts

approx, [10](#)

ecdf, [2](#)

gstat package, [1](#)

order, [4](#)

pnorm, [2](#)

points, [2](#)

qqline, [3](#)

qqnorm, [3](#)

seq, [2](#)

sp package, [1](#)

