

Spatial Statistics

Computer lab – Session 2

Madlene Nussbaum

Andreas Papritz (ehem. ETHZ)

15 Oct 2024

Table of contents

1	Geostatistical analysis of elevation data (continued)	1
1.1	Maximum likelihood estimation of spatial linear model	1
1.2	Residual diagnostics	5
1.3	Comparing alternative models	7
1.4	Stepwise model selection	10
1.5	Computing kriging predictions	13
2	Influence of variogram parameters on universal kriging predictions	16

1 Geostatistical analysis of elevation data (continued)

For the following exercises use the R code of the lecture slides on the introduction into spatial data and the analysis of the Wolfcamp aquifer dataset as template.

You will continue to analyse the `elevation` data from the exercises of *session 1*.

1.1 Maximum likelihood estimation of spatial linear model

So far, you explored the data, you fitted a trend model, you plotted a sample variogram and you fitted a spherical variogram function.

Task 1

Next estimate the parameters of trend model and variogram jointly by maximum likelihood (ML). Use the same trend model and the same variogram function as before.

💡 Task 2

Compare the ML estimates of the regression coefficients and their standard errors with those obtained before by `lm()`. Compare the ML estimates of the variogram parameters with those obtained by fitting a variogram function to the sample variogram of the residuals.

💡 Task 3

Refit the sample variogram from session 1 (last task). Then, overlay the two estimated variogram functions on the sample variogram of the residuals.

💡 Task 4

Report the confidence intervals for the variogram parameters.

Solution Task 1

```
library(georob)
data(elevation)

r.georob <- georob(height~x+y+I(x^2)+I(y^2)+x:y, elevation, locations= ~x+y,
  variogram.model="RMspheric", param=c(variance=800, nugget=0.01, scale=2),
  tuning.psi=1000, control=control.georob(ml.method="ML"))
summary(r.georob)
```

```
Call:georob(formula = height ~ x + y + I(x^2) + I(y^2) + x:y, data = elevation,
  locations = ~x + y, variogram.model = "RMspheric", param = c(variance = 800,
  nugget = 0.01, scale = 2), tuning.psi = 1000, control = control.georob(ml.method = "ML"))
```

Tuning constant: 1000

Convergence in 5 function and 3 Jacobian/gradient evaluations

Estimating equations (gradient)

	eta	scale
Gradient	: -1.183047e-04	-4.794415e-02

Maximized log-likelihood: -234.5265

Predicted latent variable (B):

Min	1Q	Median	3Q	Max
-54.670	-21.913	-6.532	10.734	101.996

Residuals (epsilon):

Min	1Q	Median	3Q	Max
-1.036e-03	-3.362e-04	-8.125e-06	3.245e-04	1.320e-03

Standardized residuals:

Min	1Q	Median	3Q	Max
-2.65357	-0.63723	-0.01043	0.58384	3.16553

Gaussian ML estimates

Variogram: RMspheric

	Estimate	Lower	Upper
variance	7.631e+02	4.993e+02	1.166e+03
snugget(fixed)	0.000e+00	NA	NA
nugget	9.537e-03	1.109e-78	8.202e+73
scale	2.309e+00	1.849e+00	2.885e+00

Fixed effects coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	956.41685	27.58022	34.678	< 2e-16 ***
x	-48.02765	12.81776	-3.747	0.000498 ***
y	-18.33591	12.13964	-1.510	0.137777
I(x^2)	6.76066	1.68171	4.020	0.000214 ***
I(y^2)	-0.27943	1.64189	-0.170	0.865610
x:y	-0.09142	1.51417	-0.060	0.952118

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error (sqrt(nugget)): 0.09766

Robustness weights:

All 52 weights are ~= 1.

Solution Task 2

The ML estimates of the regression coefficients differ mostly for y , y^2 and $x:y$ from the estimates by `lm()`. The standard errors of the ML estimates are again larger. Only the coefficients for x and x^2 differ significantly from zero.

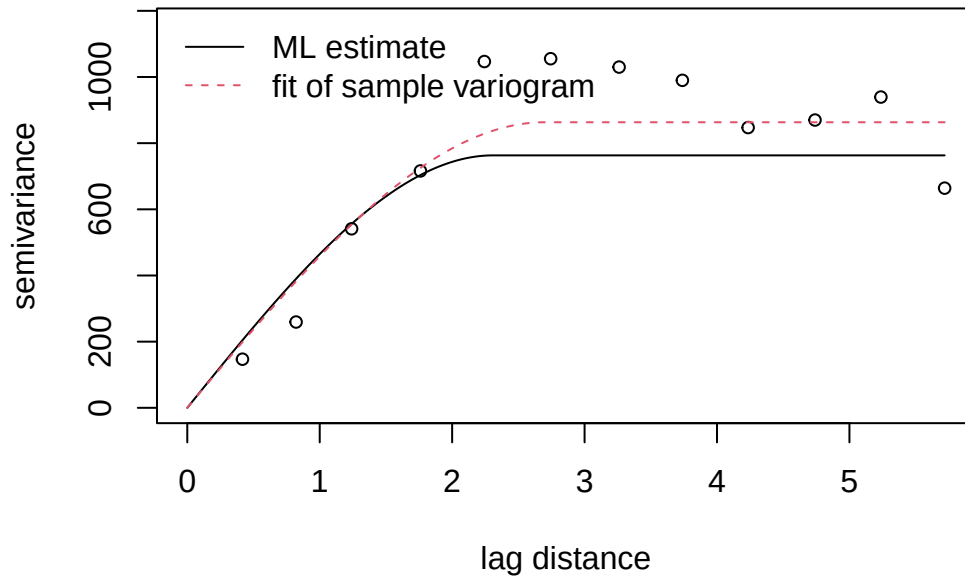
The estimated nugget is still close to zero and the estimated sill and range parameters are about the same as the values obtained by fitting a spherical variogram function to the sample variogram.

Solution Task 3

```
# linear model, sample variogram and variogram fit (session 1)
r.lm <- lm(height~x+y+I(x^2)+I(y^2)+x:y, elevation)
elevation$res <- residuals(r.lm)

r.v <- sample.variogram(elevation$res, locations = elevation[, c("x", "y")],
  estimator = "matheron", lag.dist.def = 0.5, max.lag = 6)
r.v.sph <- fit.variogram.model(r.v, variogram.model="RMspheric",
  param=c(variance=700, nugget=0.01, scale=2), max.lag=5)

plot(r.georob, lag.dist.def=0.5, estimator="matheron", max.lag=6)
lines(r.v.sph, col=2, lty=2)
legend("topleft", lty=1:2, col=1:2, bty="n",
  legend=c("ML estimate", "fit of sample variogram"))
```



Solution Task 4

The estimated nugget has an extremely wide 95% confidence interval! The confidence intervals for these parameters are equal to $[500, 1166]$ (sill) and $[1.85, 2.89]$ (range).

1.2 Residual diagnostics

💡 Task 1

Assess the ML fit of the spatial model by a Tukey-Anscombe plot and a normal QQ-plot of the ML regression residuals $Y_i - \mathbf{d}(\mathbf{x}_i)^T \hat{\boldsymbol{\beta}} = \hat{E}(\mathbf{x}_i) + \hat{Z}_i$.

💡 Task 2

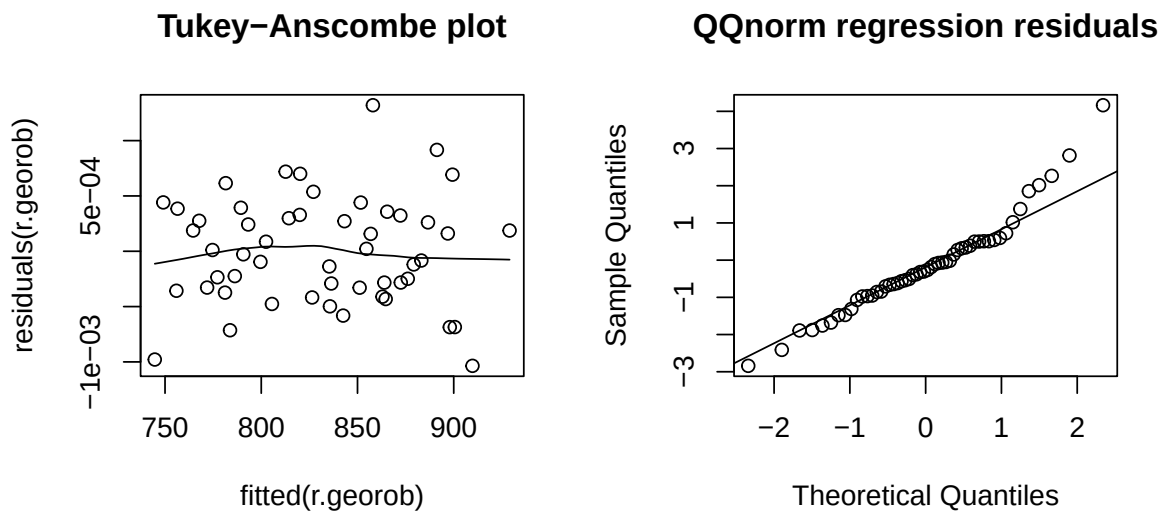
Create in addition a normal QQ-plot of the estimated independent errors $\hat{Z}_i$ and of the spatially correlated component $\hat{E}(\mathbf{x}_i)$ of the residuals. Comment on the plots.

Hint

Use `?residuals.georob`, `?rstandard.georob` and `?ranef.georob` to see how to extract the various types of residuals from the fitted `georob` object.

Solution Task 1

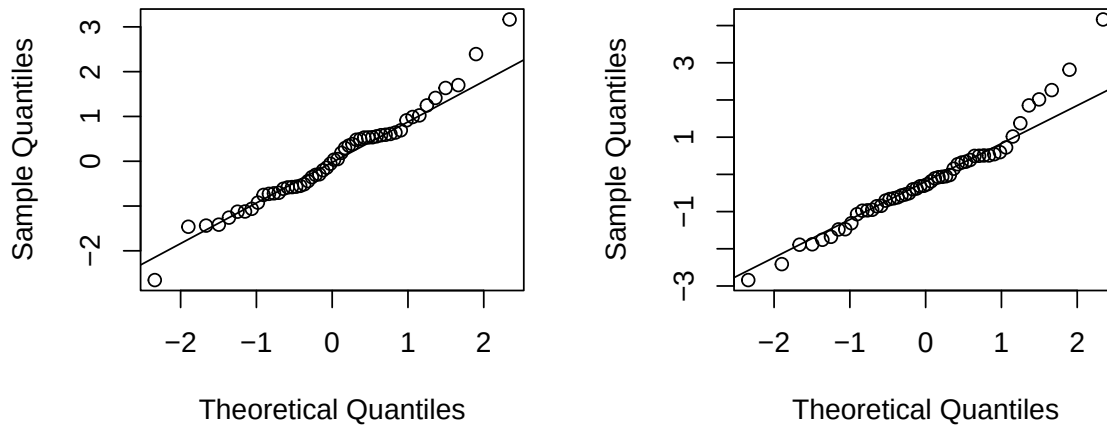
```
par(mfrow=c(1, 2))
scatter.smooth(fitted(r.georob), residuals(r.georob), main="Tukey-Anscombe plot")
qqnorm(rstandard(r.georob, level=0), main="QQnorm regression residuals")
qqline(rstandard(r.georob, level=0))
```



Solution Task 2

```
par(mfrow=c(1, 2))
qqnorm(rstandard(r.georob, level=1), main="QQnorm estimated indep. errors")
qqline(rstandard(r.georob, level=1))
qqnorm(ranef(r.georob, standard=TRUE), main="QQnorm estimated auto-correlated errors")
qqline(ranef(r.georob, standard=TRUE))
```

QQnorm estimated indep. errors QQnorm estimated auto-correlated er



The diagnostic plots of the regression residuals show a heavier upper tail. Otherwise there are no gross violations of the modelling assumptions. Note that the normal QQ-plots of the regression residuals and of the spatially auto-correlation error components are very similar because the estimated independent error components are very small (nugget (= variance of Z_i) is very small).

1.3 Comparing alternative models

💡 Task 1

If you have not yet done so, fit now a spatial model that uses a quadratic function in the coordinates (second order polynomial) as trend model. Use again ML for estimating the regression coefficients and the variogram parameters.

💡 Task 2

Fit in addition a model that uses a linear function of the coordinates (first-order polynomial) as trend model.

💡 Task 3

Use a Wald test to compare the fit of the two models. Compare also the fitted variogram parameters of the two models.

Which model fits the elevation data better?

Solution Task 1

See solution Task 1 in section 1.1.

Solution Task 2

```
r.georob.lin <- update(r.georob, ~x+y)
summary(r.georob.lin)
```

```
Call:georob(formula = height ~ x + y, data = elevation, locations = ~x +
  y, variogram.model = "RMspheric", param = c(variance = 800,
  nugget = 0.01, scale = 2), tuning.psi = 1000, control = control.georob(ml.method = "ML"))
```

Tuning constant: 1000

Convergence in 7 function and 5 Jacobian/gradient evaluations

Estimating equations (gradient)

	eta	scale
Gradient	: -2.721740e-04	-8.779291e-02

Maximized log-likelihood: -241.6297

Predicted latent variable (B):

Min	1Q	Median	3Q	Max
-129.8469	-46.2072	-18.2202	0.1406	85.9510

Residuals (epsilon):

Min	1Q	Median	3Q	Max
-3.105e-03	-1.260e-03	5.885e-05	9.783e-04	3.929e-03

Standardized residuals:

Min	1Q	Median	3Q	Max
-1.69918	-0.78690	0.03026	0.65151	3.05067

Gaussian ML estimates

Variogram: RMspheric

	Estimate	Lower	Upper
variance	2.425e+03	1.598e+03	3.679e+03
snugget(fixed)	0.000e+00	NA	NA
nugget	3.030e-02	1.112e-53	8.258e+49
scale	6.199e+00	5.217e+00	7.366e+00

Fixed effects coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	909.281	42.826	21.232	<2e-16 ***
x	-6.518	7.831	-0.832	0.409
y	-10.640	7.723	-1.378	0.175

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error (sqrt(nugget)): 0.1741

Robustness weights:

All 52 weights are ~= 1.

Solution Task 3

```
waldtest(r.georob, r.georob.lin, test="F")
```

Wald test

Model 1: height ~ x + y + I(x^2) + I(y^2) + x:y

Model 2: height ~ x + y

	Res.Df	Df	F	Pr(>F)
1	46			
2	49	-3	5.5028	0.002579 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The estimated sill and range parameters are larger for the linear trend function model and the maximized log-likelihood decreased substantially (-241.6 vs. -234.5). This indicates that the model with the simpler trend function underfits the data. This is confirmed by the Wald test.

1.4 Stepwise model selection

💡 Task 1

Then use stepwise backward and forward covariate selection by `step()` to simplify the model. Use the argument `fixed.add1.drop1=FALSE` when using `step()`. This has the effect that the variogram parameters are re-fitted for each tested model. What model minimizes the AIC?

💡 Task 2

Compare the three models by 10-fold cross-validation. Which model do you prefer based on bias (`me`) and root mean square error (`rmse`)?

Solution Task 1

```
step(r.georob, fixed.add1.drop1=FALSE)
```

Start: AIC=487.05

```
height ~ x + y + I(x^2) + I(y^2) + x:y
```

	Df	AIC	Converged
- x:y	1	485.06	1
- I(y^2)	1	485.08	1
<none>		487.05	1
- I(x^2)	1	498.92	1

Step: AIC=485.06

```
height ~ x + y + I(x^2) + I(y^2)
```

	Df	AIC	Converged
- I(y^2)	1	483.08	1
<none>		485.06	1
- y	1	485.63	1
+ x:y	1	487.05	1
- I(x^2)	1	496.94	1
- x	1	497.63	1

Step: AIC=483.08

```
height ~ x + y + I(x^2)
```

	Df	AIC	Converged
<none>		483.08	1
+ I(y^2)	1	485.06	1
+ x:y	1	485.08	1
- y	1	490.74	1
- I(x^2)	1	495.26	1
- x	1	495.93	1

Tuning constant: 1000

Fixed effects coefficients:

(Intercept)	x	y	I(x^2)
959.081	-48.465	-20.414	6.786

Variogram: RMspheric

variance	snugget(fixed)	nugget	scale
7.617e+02	0.000e+00	9.519e-03	2.303e+00

```
r.georob.aic <- update(r.georob, . ~ x + y + I(x^2))
summary(r.georob.aic)
```

Call: georob(formula = height ~ x + y + I(x^2), data = elevation, locations = ~x + y, variogram.model = "RMspheric", param = c(variance = 800, nugget = 0.01, scale = 2), tuning.psi = 1000, control = control.georob(ml.method = "ML"))

Tuning constant: 1000

Convergence in 5 function and 3 Jacobian/gradient evaluations

Estimating equations (gradient)

	eta	scale
Gradient	: -1.134449e-04	-2.057632e-02

Maximized log-likelihood: -234.5419

Predicted latent variable (B):

Min	1Q	Median	3Q	Max
-55.978	-20.956	-6.443	10.837	101.884

Residuals (epsilon):

Min	1Q	Median	3Q	Max
-1.043e-03	-3.336e-04	-1.752e-05	3.260e-04	1.315e-03

Standardized residuals:

Min	1Q	Median	3Q	Max
-2.46462	-0.63532	-0.04007	0.58487	3.15347

Gaussian ML estimates

Variogram: RMspheric

	Estimate	Lower	Upper
variance	7.616e+02	4.979e+02	1.165e+03
snugget(fixed)	0.000e+00	NA	NA
nugget	9.518e-03	2.088e-80	4.339e+75
scale	2.302e+00	1.839e+00	2.882e+00

Fixed effects coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	959.085	19.346	49.575	< 2e-16 ***
x	-48.464	11.399	-4.252	9.74e-05 ***
y	-20.415	3.154	-6.474	4.72e-08 ***
I(x^2)	6.786	1.669	4.066	0.000177 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error (sqrt(nugget)): 0.09756

Robustness weights:

All 52 weights are ~ = 1.

The terms x, x^2 and y are kept for the minimum AIC model, which again suggests that a (reduced) quadratic trend model is better.

Remark: Note that it is generally not a good idea to eliminate some “non-significant” terms of a higher order polynomial trend function unless you are working with orthogonal polynomials. When you drop some terms then a shift of the origin of the coordinates usually affects the goodness-of-fit of the trend model strongly. If all terms are present, then the goodness-of-fit is invariant to a shift of the spatial coordinates.

Solution Task 2

```
r.cv <- cv(r.georob, seed=735)
r.cv.lin <- georob::cv(r.georob.lin, seed=735)
r.cv.aic <- georob::cv(r.georob.aic, seed=735)
summary(r.cv)
```

Statistics of cross-validation prediction errors

me	mede	rmse	made	qne	msse	medsse	crps
-7.9471	-11.3857	34.4851	31.2025	26.0014	1.2489	0.4723	18.3142

```
summary(r.cv.lin)
```

Statistics of cross-validation prediction errors

me	mede	rmse	made	qne	msse	medsse	crps
-7.3266	-8.3039	35.7196	35.7881	34.5415	1.3350	0.7425	19.6755

```
summary(r.cv.aic)
```

Statistics of cross-validation prediction errors

me	mede	rmse	made	qne	msse	medsse	crps
-4.1330	-3.3323	26.4021	21.9314	23.4301	1.0352	0.3756	14.4114

The mes of the models hardly differ. However, the root mean square errors **rmse** (and their robust variants **made** and **qne**) indicate that the reduced quadratic trend model selected by **step()** performs somewhat better.

1.5 Computing kriging predictions

Task 1

Using the model which minimized AIC in the previous problem, compute the kriging predictions of **height** for a rectangle with the lower left corner in (0.1, -0.1) and top right corner in (6.4, 6.3). Use a grid increment of 0.05 when you generate the nodes of the grid for which you compute the predictions.

Convert the dataframe with the prediction results to a **SpatRaster** using the functions **rast** of the package **terra**.

💡 Task 2

Plot the predicted **height** and its prediction standard error along with the lower and upper limits of pointwise 95%-prediction intervals.

💡 Task 3

For comparison, we plot also the quadratic trend surface along with the bubble plot of the regression residuals superimposed. Use the **points** second level plot function to add the bubbles.

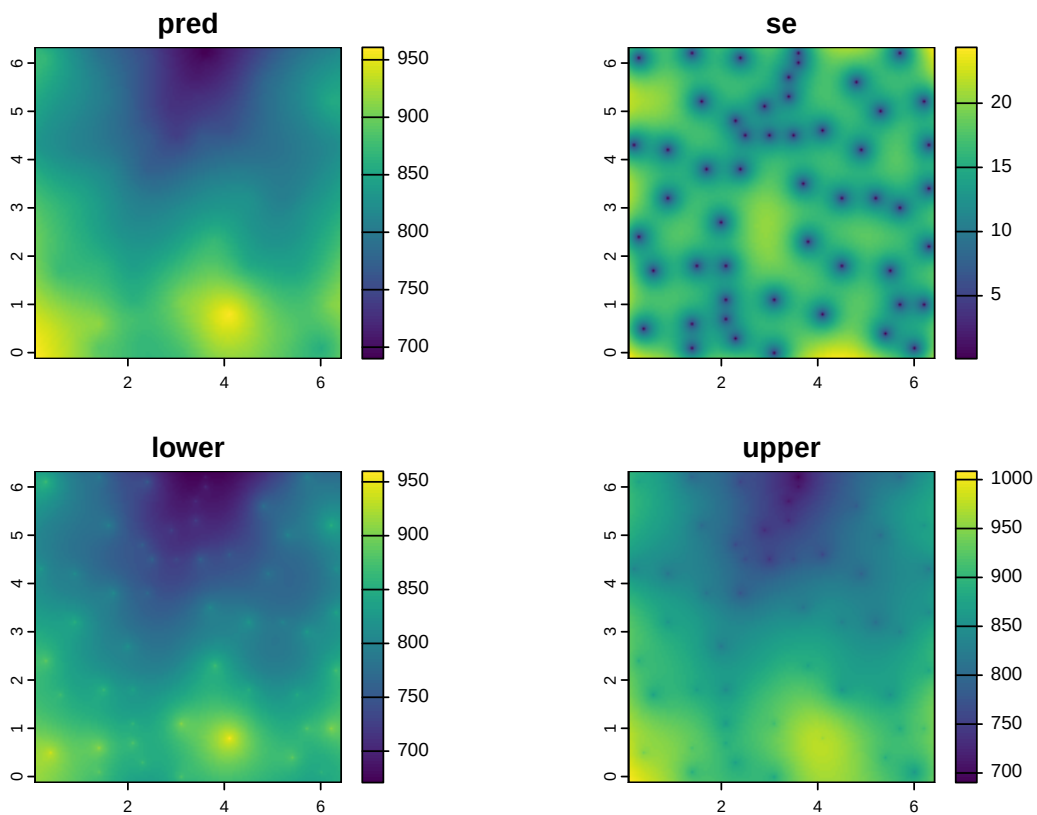
Solution Task 1

```
library(terra)
elevation.grid <- expand.grid(
  x=seq(0.1, 6.4, by=0.05),
  y=seq(-0.1, 6.3, by=0.05)
)
r.uk <- predict(r.georob.aic, newdata=elevation.grid)
r.uk <- rast(r.uk)
r.uk
```

```
class      : SpatRaster
dimensions : 129, 127, 4  (nrow, ncol, nlyr)
resolution : 0.05, 0.05  (x, y)
extent     : 0.075, 6.425, -0.125, 6.325  (xmin, xmax, ymin, ymax)
coord. ref.:
source(s)  : memory
names      :      pred,          se,      lower,      upper
min values : 690.0010,  0.09755892, 670.8158,  690.1922
max values : 960.8112, 24.35006676, 959.8075, 1008.5365
```

Solution Task 2

```
plot(r.uk)
```

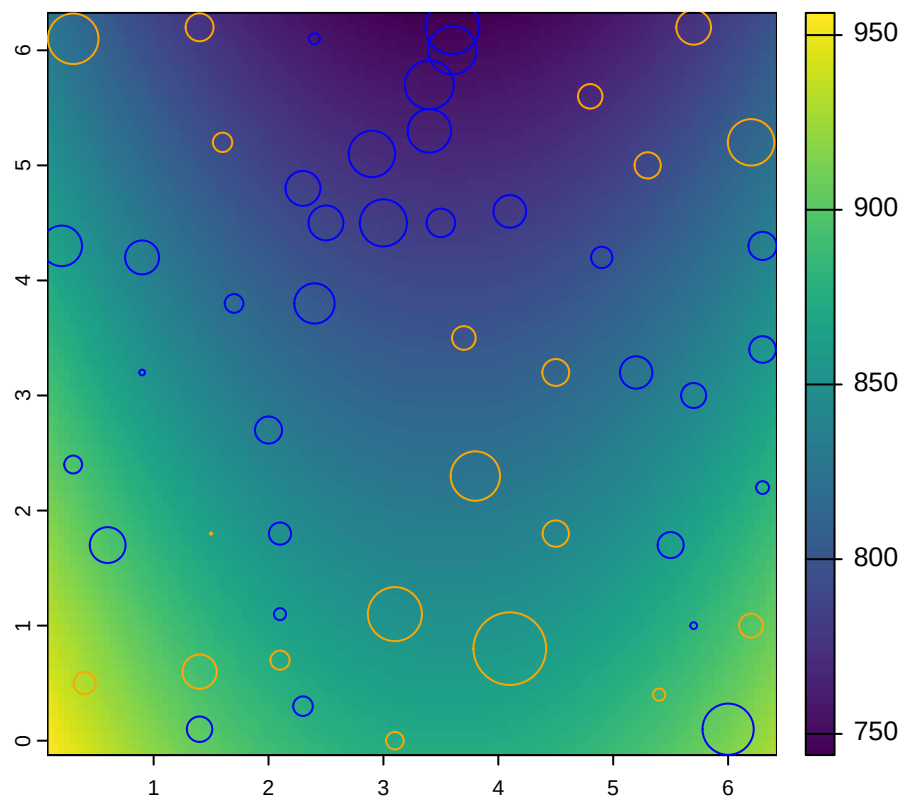


Solution Task 3

```
r.ts <- predict(r.georob.aic, newdata=elevation.grid, type="trend")
r.ts <- rast(r.ts)

plot(r.ts[["pred"]])

res <- residuals(r.georob.aic, level=0)
points(x=elevation$x, y=elevation$y,
       cex=sqrt(abs(res))/2,
       col=c("blue", NA, "orange")[sign(res)+2])
```



When comparing the kriging with the trend surface predictions, note how the kriging surface “follows” the residuals of the regression model.

2 Influence of variogram parameters on universal kriging predictions

Load the `geoR` package which provides the `wolfcamp` data. Create by the following code a `data.frame` that contains a grid of prediction points.

```
library(geoR)
d.wolfcamp.grid <- expand.grid(
  x=seq(from=-250, to=250, length=52),
  y=seq(from=-150, to=150, length=32)
)
```


Then use the following code to explore the effect of the variogram parameters on the UK predictions:

```
persp(
  krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
    krige=krige.control(trend.d="1st", trend.l="1st",
      cov.model="spherical", cov.pars=c(sill=1000, range=100),
      nugget=3000)),
  theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure"
)
```

💡 Task 1

Vary the nugget-to-sill ratio for a total sill (= nugget + sill) equal to 4000.

💡 Task 2

Using a constant nugget-to-sill ratio of 1:3 and a range equal to 100, compare the predictions for a total sill of 400 and 4000.

💡 Task 3

Vary the range for a nugget-to-sill ratio equal to 1:3.

Solution Task 1

For the nugget-to-sill ratio we create the plots for the values 0:4, 1:3, 2:2, 39:1.

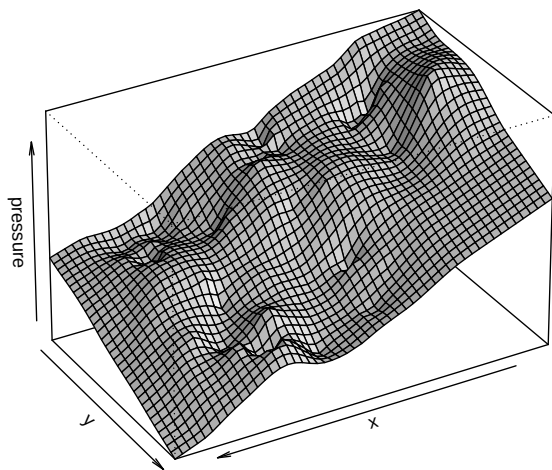
```
par(mfrow=c(2,2), mar=c(0, 2, 2, 0))
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=4000, range=100),
    nugget=0)), theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
  main="nugget:sill = 0:4")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=3000, range=100),
    nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
```

```

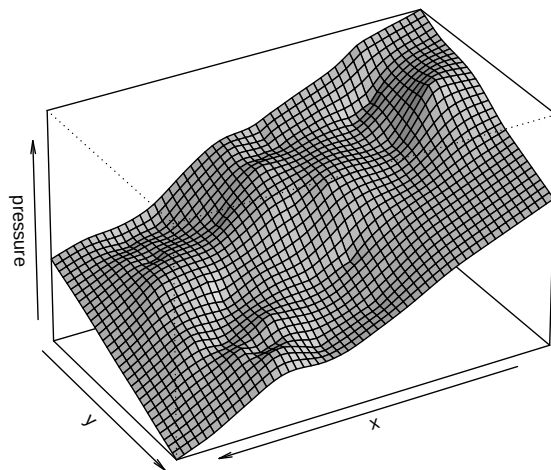
    main="nugget:sill = 1:3")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
    krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=2000, range=100),
    nugget=2000)), theta=150, phi=30, d=5, scale=FALSE,
    expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
    main="nugget:sill = 2:2")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
    krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=100, range=100),
    nugget=3900)), theta=150, phi=30, d=5, scale=FALSE,
    expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
    main="nugget:sill = 39:1")

```

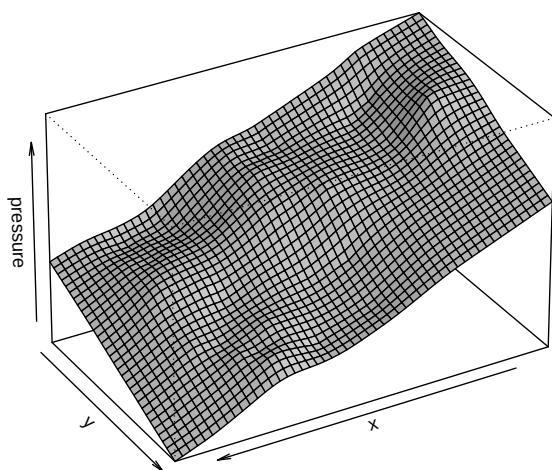
nugget:sill = 0:4



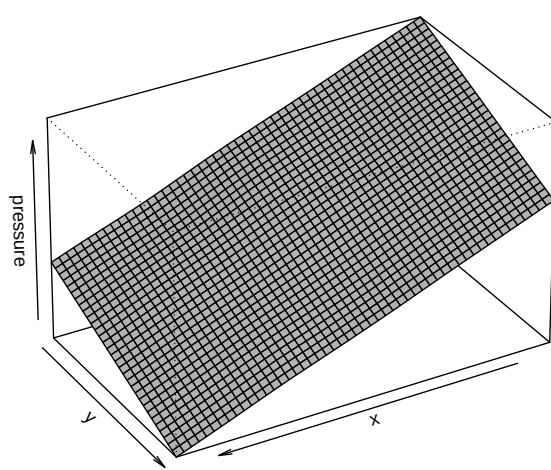
nugget:sill = 1:3



nugget:sill = 2:2



nugget:sill = 39:1



The nugget-to-sill ratio determines how strongly kriging smooths the data. For zero nugget, kriging is an exact interpolator that predicts at the data locations $\mathbf{x}_i$ the observations y_i (which are then equal to the signal $S(\mathbf{x}_i)$). Of course, the surface of the kriging predictions is then everywhere continuous.

For non-zero nugget kriging is no longer an exact interpolator (except when we predict the response variable Y ; the kriging surface then has discontinuities at the data locations, see slides).

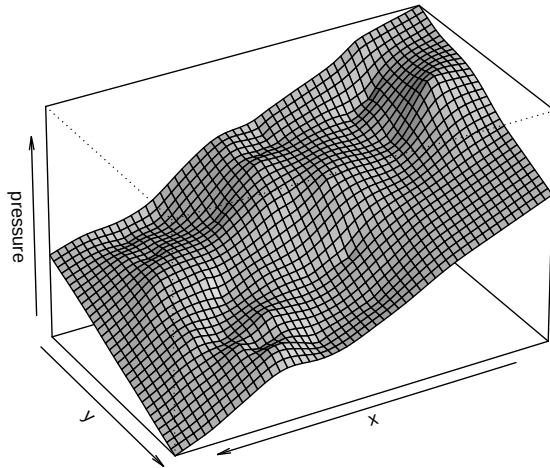
The larger the nugget-to-sill ratio the more strongly kriging smooths the data: For increasing

nugget-to-sill ratio the simple kriging weights decrease (because the elements of $\mathbf{s}_Y$ are becoming small), hence the contribution of $(Y - \hat{Y}_{GLS})$ to the UK prediction becomes small as well, and vanishes for a pure nugget variogram completely so that $\hat{S}_k = \hat{S}_{OLS}$, i.e. UK is equal to trend surface prediction (remember: for a pure nugget variogram $\hat{\beta}_{GLS} = \hat{\beta}_{OLS}$).

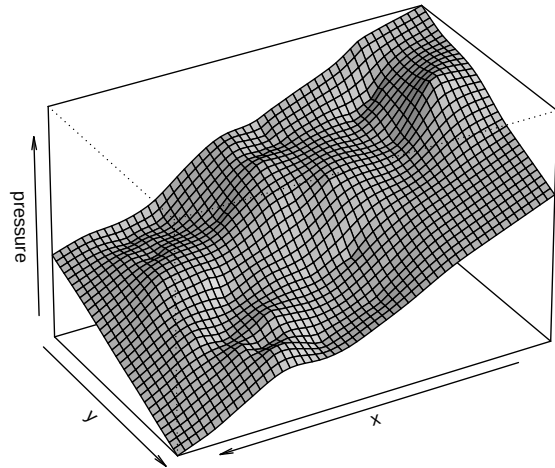
Solution Task 2

```
par(mfrow=c(2,2), mar=c(0, 2, 2, 0))
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=3000, range=100),
    nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
  main="UK prediction total sill 4000")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=300, range=100),
    nugget=100)), theta=150, phi=30, d=5, scale=FALSE,
  expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
  main="UK prediction total sill 400")
persp(tmp <- krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=3000, range=100),
    nugget=1000)), values=sqrt(tmp$krige.var), theta=150, phi=30,
  d=5, scale=FALSE, expand=6, shade=0.5, xlab="x", ylab="y",
  zlab="pressure", main="UK standard error total sill 4000")
persp(tmp <- krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
  krige=krige.control(trend.d="1st", trend.l="1st",
    cov.model="spherical", cov.pars=c(sill=300, range=100),
    nugget=100)), values=sqrt(tmp$krige.var), theta=150, phi=30,
  d=5, scale=FALSE, expand=6, shade=0.5, xlab="x", ylab="y",
  zlab="pressure", main="UK standard error total sill 400")
```

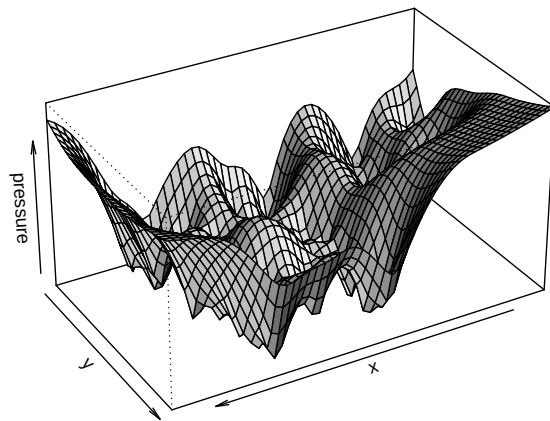
UK prediction total sill 4000



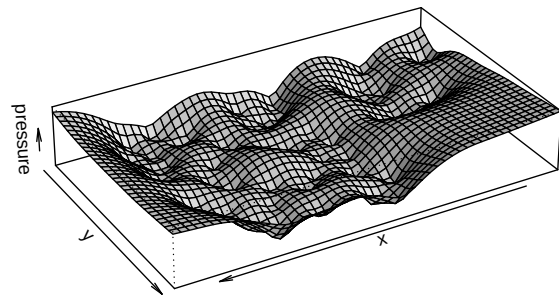
UK prediction total sill 400



UK standard error total sill 4000



UK standard error total sill 400



Unlike the prediction standard error, the kriging predictions do not depend on the total sill of the variogram because the simple kriging weights $\mathbf{S} \mathbf{Y}^{-1}$ are a function of the autocorrelogram only.

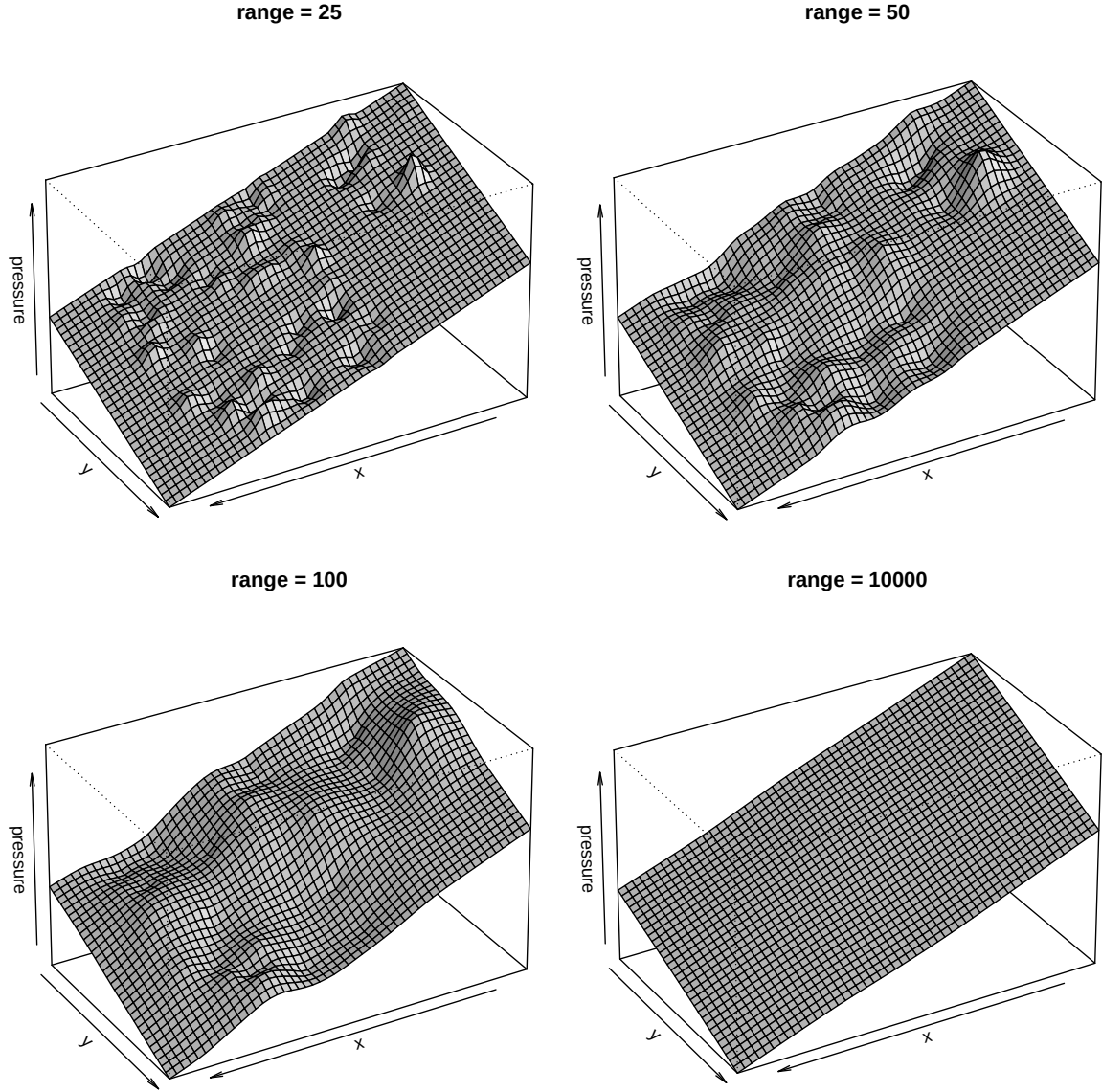
Solution Task 3

```
par(mfrow=c(2,2), mar=c(0, 2, 2, 0))
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
```

```

krige=krige.control(trend.d="1st", trend.l="1st",
cov.model="spherical", cov.pars=c(sill=3000, range=25),
nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
main="range = 25")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
krige=krige.control(trend.d="1st", trend.l="1st",
cov.model="spherical", cov.pars=c(sill=3000, range=50),
nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
main="range = 50")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
krige=krige.control(trend.d="1st", trend.l="1st",
cov.model="spherical", cov.pars=c(sill=3000, range=100),
nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
main="range = 100")
persp(krige.conv(geodata=wolfcamp, locations=d.wolfcamp.grid,
krige=krige.control(trend.d="1st", trend.l="1st",
cov.model="spherical", cov.pars=c(sill=3000, range=10000),
nugget=1000)), theta=150, phi=30, d=5, scale=FALSE,
expand=0.3, shade=0.5, xlab="x", ylab="y", zlab="pressure",
main="range = 10000")

```



The range controls “how far” the local adjustments $(\mathbf{Y} - \mathbf{X}\hat{\mathbf{Y}}_{\text{GLS}})$ of the trend surface prediction surface extend around the data locations. For a small range, we see isolated “peaks” and “funnels” that mostly do not overlap except for the zones where there are multiple observations at very short distance from each other. For larger range, the local adjustments overlap at most points within the prediction domain and one can no longer identify the contributions of single data points.

For very large range and non-negligible nugget-to-sill ratio the local adjustments extend very far so that at all prediction points the contributions of all data points matter. The prediction

surface is then again very close to the trend surface because on average the local adjustments cancel out. Note that this is not true when there is no nugget. Then—as kriging is then an exact interpolator—the kriging surface is forced to go through all observations. Hence the UK predictor must vary locally in the zone where there are data points (check this!).