

# Intro to geostatistical analysis

## Spatial Statistics

Madlene Nussbaum

[m.nussbaum@uu.nl](mailto:m.nussbaum@uu.nl)

Utrecht University, Department of Physical Geography

# 1 Geostatistical model notation

## Objectives of geostatistical analysis

1. Computing spatial **predictions** of response variable for
  - a location without measurement, i.e. a creating spatially continuous surface
  - finite support targets (spatial means)
2. Quantifying **uncertainty** of these predictions
3. Estimation of **parameters** of (regression) models fitted to geostatistical data

# Geostatistical data sets

- Set of observations  $(y_i, \mathbf{x}_i)$  where  $y_i$  is a datum of a response variable and  $\mathbf{x}_i$  is a spatial location in a study domain  $D$
- Optional: spatial covariates, say  $d_k(\mathbf{x}_i)$ , used to “explain” the spatial pattern of the response variable
- Geostatistical data often show (gradual) large-scale spatial variation and small-scale local fluctuations
  - **trend**: commonly modelled as low-order polynomial function of spatial coordinates ( $\Rightarrow$  trend surface) or as function of external spatial covariates ( $\Rightarrow$  external trend)
  - **local fluctuations** usually spatially structured (values at pairs of nearby locations “more similar” than for pairs farther apart)  $\Rightarrow$  spatial auto-correlation
- Spatial data is sometimes distorted by independent measurement errors

# Terminology and model notation

Model for data:  $Y_i = S(\mathbf{x}_i) + Z_i$

where

$Y_i$  :  $i^{\text{th}}$  datum

$S(\mathbf{x}_i)$  : "signal" (= true quantity) at location  $\mathbf{x}_i$

$Z_i$  : iid. random measurement error

Decomposition of signal into trend  $\mu(\mathbf{x}_i)$  and stochastic fluctuation:

$$S(\mathbf{x}_i) = \mu(\mathbf{x}_i) + E(\mathbf{x}_i)$$

where commonly a linear model is used for  $\mu(\mathbf{x}_i)$

$$\mu(\mathbf{x}_i) = \sum_k d_k(\mathbf{x}_i) \beta_k = \mathbf{d}(\mathbf{x}_i)^T \boldsymbol{\beta}$$

with  $d_k(\mathbf{x}_i)$  denoting (spatial) covariates and  $\{E(\mathbf{x}_i)\}$  a zero mean stochastic process (random field).

# In summary

Geostatistical data  $(y_i, \mathbf{x}_i, d_k(\mathbf{x}_i))$ :

1. response  $y_i$
2. location  $\mathbf{x}_i$
3. covariates  $d_k(\mathbf{x}_i)$
4. often approximately infinitesimal support

Models for geostatistical data decompose spatial variation (non-uniquely) into “large-scale” trend and local auto-correlated fluctuations

- trend modelled by linear regression model
- local fluctuations modelled by auto-correlated stochastic process

# 2 Example data: Wolfcamp aquifer

## Wolfcamp aquifer data set

- Data on hydraulic head (pressure) in a confined brine aquifer in NW Texas
- Hydro-geological study part of evaluation whether region suited to host nuclear waste repository
- Measurement of hydraulic head at 85 locations
- Data set part of R packages *georob* and *geoR*

```
1 library(sp)
2 library(georob)
3 data(wolfcamp, package="georob")
4 str(wolfcamp)
```

```
'data.frame': 85 obs. of 3 variables:
 $ x      : num  68.85 -44.09 -1.87 -29.96 155.24 ...
 $ y      : num  44.5 -14.8 -24.3 -37.9 -57 ...
 $ pressure: num  446 778 658 748 535 ...
```

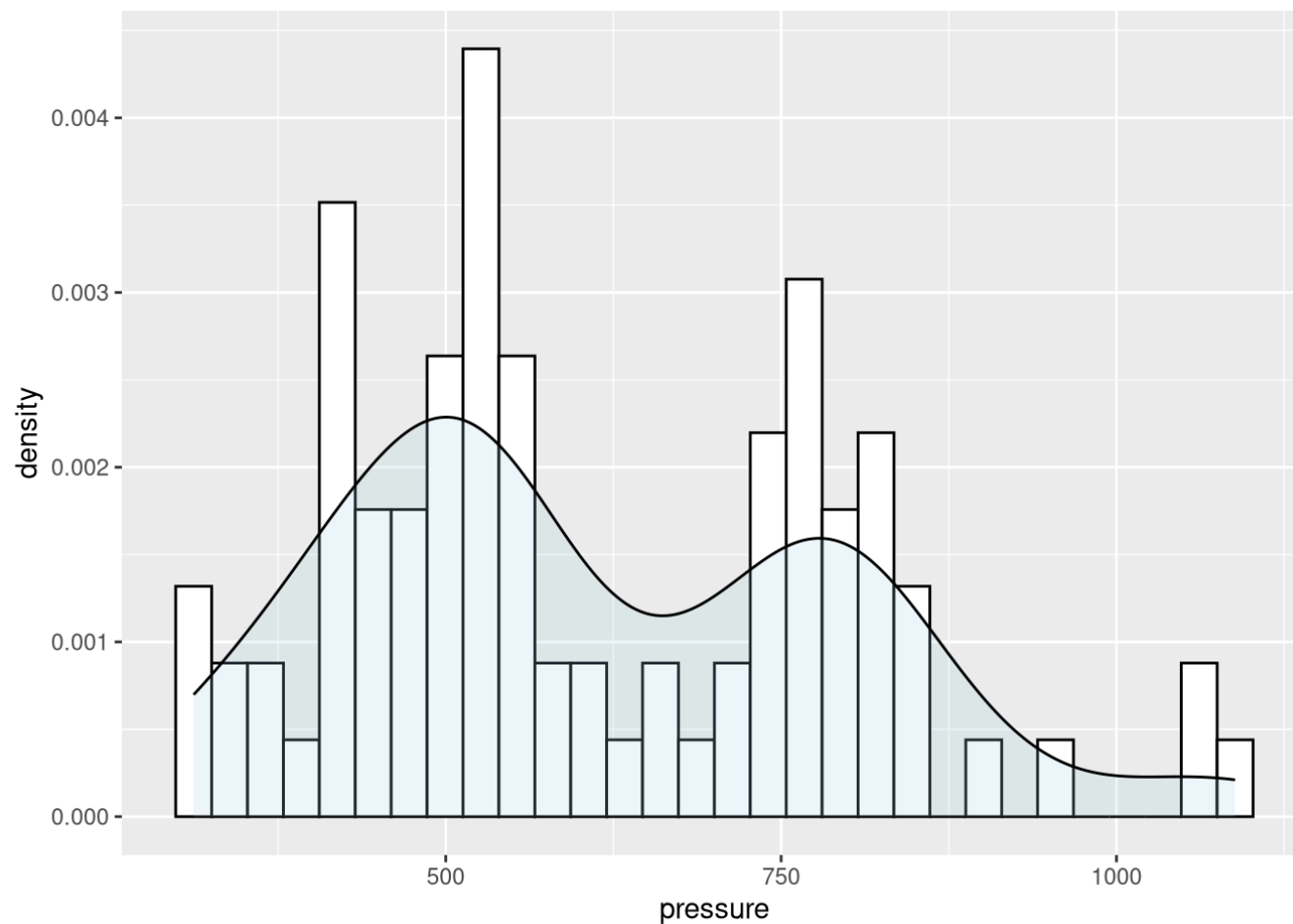
# Exploratory analysis

```
1 # create sf object
2 library(sf)
3 d.w <- st_as_sf(wolfcamp, coords = c("x", "y"))
4 summary(d.w)
```

| pressure       | geometry   |
|----------------|------------|
| Min. : 312.1   | POINT :85  |
| 1st Qu.: 471.8 | epsg:NA: 0 |
| Median : 547.7 |            |
| Mean : 610.3   |            |
| 3rd Qu.: 774.2 |            |
| Max. :1088.4   |            |

# Distribution of response

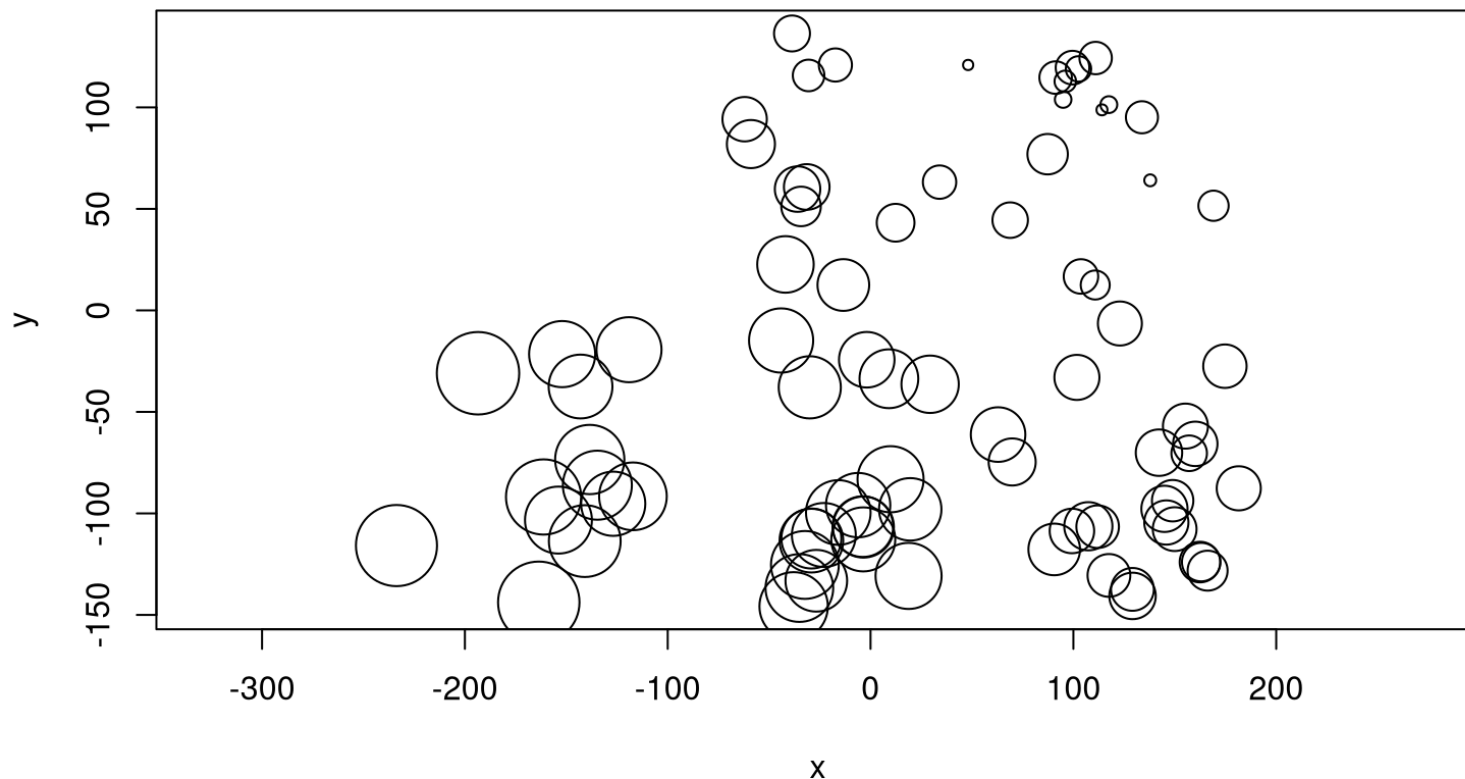
```
1 library(ggplot2)
2 ggplot(wolfcamp, aes(x = pressure)) +
3   geom_histogram(aes(y=..density..), colour="black", fill="white")+
4   geom_density(alpha=.2, fill="lightblue")
```



# Spatial distribution of values

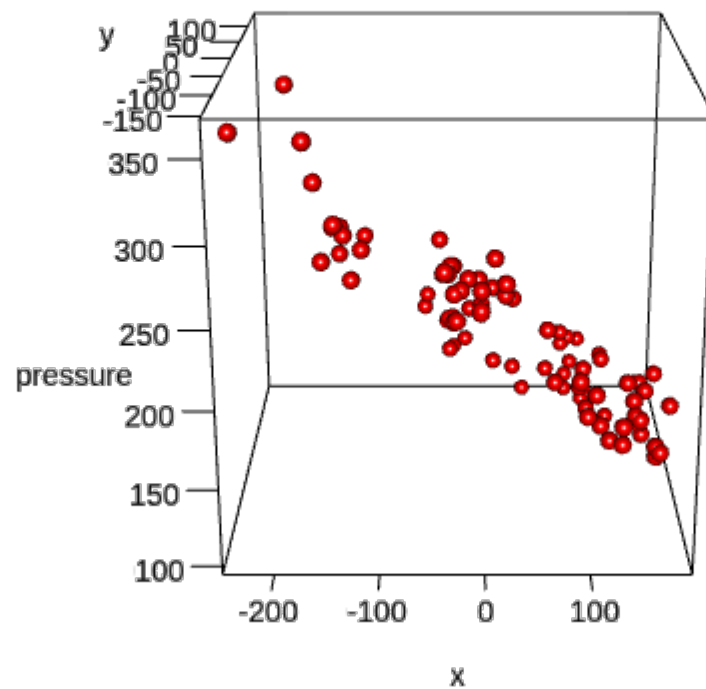
Bubble plot: symbol area linearly related to data

```
1 plot(y~x, data=wolfcamp, asp=1, cex=sqrt(pressure-300)/5)
```



# Spatial distribution of values

```
1 library(rgl)
2 plot3d(x=wolfcamp$x, y=wolfcamp$y, z=wolfcamp$pressure/3,
3        type="s", radius=7, col="red",
4        xlab="x", ylab="y", zlab="pressure")
5 rglwidget()
```



# 3 Ad-hoc: Inverse distance weighting

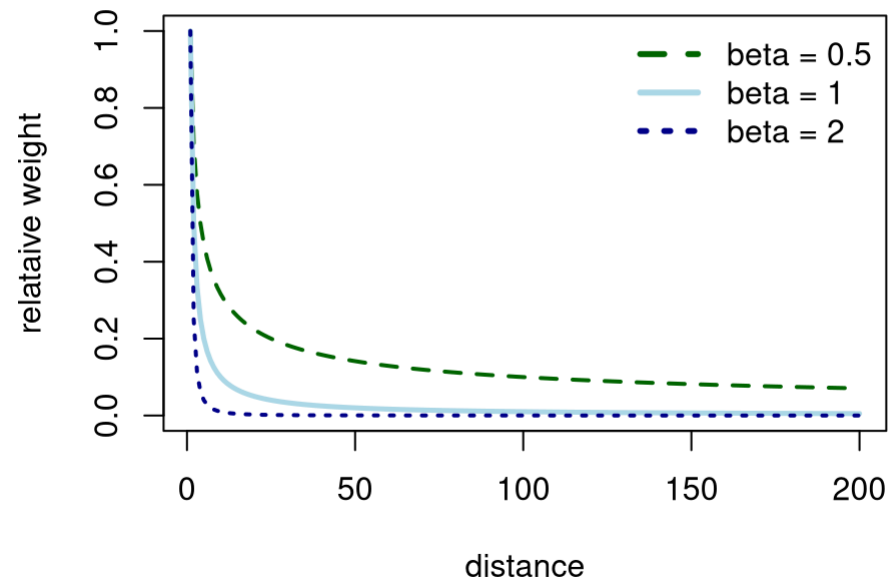
Calculate prediction  $S(\mathbf{x}_0)$  for unsampled location  $x_0$  as weighted mean of the sampling points  $x_i$  with weights  $w(x_i)$  being derived from the pairwise distances  $d(x_p, x_i)$  raised to the power  $\beta$ :

$$w_i(\mathbf{x}_0) = \frac{1}{d(x_0, x_i)^\beta}$$

With predictions  $S(\mathbf{x}_0)$ :

$$\hat{S}(\mathbf{x}_0) = \frac{\sum_{i=1}^n w_i(\mathbf{x}_0) y(\mathbf{x}_i)}{\sum_{i=1}^n w_i(\mathbf{x}_0)}$$

Decay of weights depending on beta



# Inverse distance weighting implementation

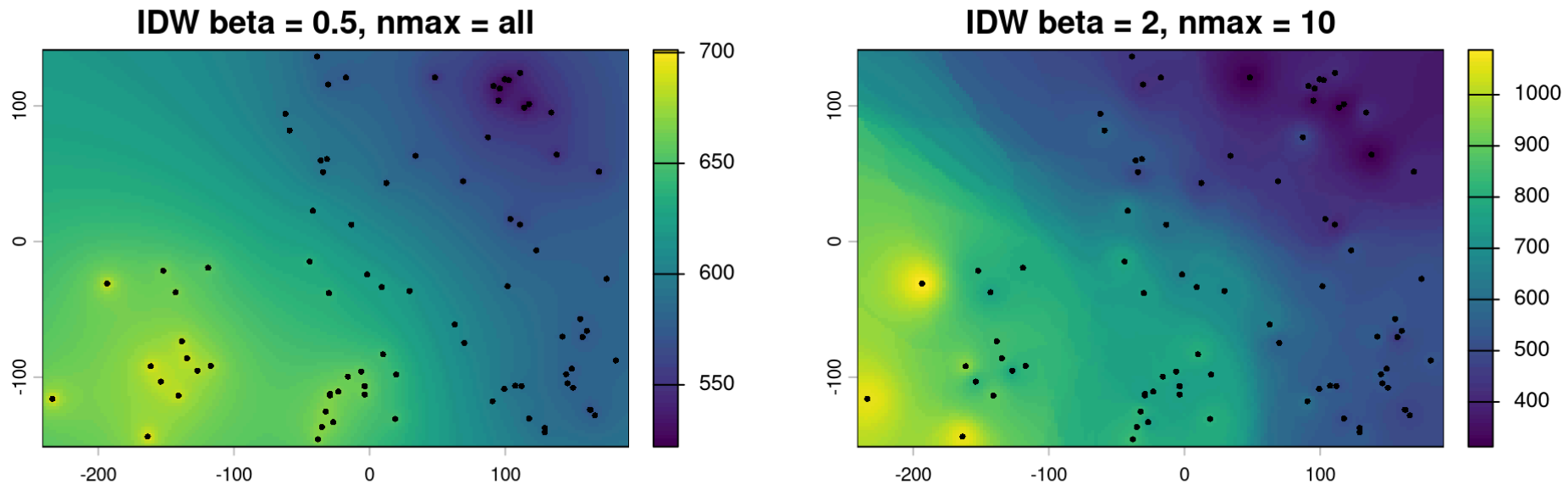
```
1 library(gstat)
2 library(terra)
3 idw <- gstat(id = "pressure", formula = pressure ~ 1, data = d.w,
4             nmax = nrow(d.w), # use all the neighbors locations
5             set = list(idp = 0.5)) # decay of weights by d^0.5
6
7 # create a regular grid
8 d.w.grid <- expand.grid(
9   x = seq(-240, 190, by= 2.5),
10  y = seq(-150, 140, by= 2.5)
11 )
12 gridded(d.w.grid) <- ~x+y
13
14 # predict for each node of grid
15 d.w.idw <- predict(idw, newdata= d.w.grid)
```

[inverse distance weighted interpolation]

```
1 ( d.w.idw <- rast(d.w.idw) )
```

```
class       : SpatRaster
size        : 117, 173, 2  (nrow, ncol, nlyr)
resolution  : 2.5, 2.5  (x, y)
extent      : -241.25, 191.25, -151.25, 141.25  (xmin, xmax, ymin, ymax)
coord. ref. :
source(s)   : memory
names       : pressure.pred, pressure.var
min values  :      522.0596,      NaN
max values  :      701.1927,      NaN
```

```
1 par(mfrow=c(1,2))
2 plot(d.w.idw["pressure.pred"], main = "IDW beta = 0.5, nmax = all"); points(d.w, cex = 0.5)
3 plot(d.w.idw2["pressure.pred"], main = "IDW beta = 2, nmax = 10"); points(d.w, cex = 0.5)
```



**Once we found optimal  $\beta$ , is our job done?!**

# Discuss ...

- What is the problem with IDW?
- How could you improve the approach?
- What would you do as next step?

# 4 Model the spatial trend

```
1 r.lm.1 <- lm(pressure~x+y, wolfcamp)
2 summary(r.lm.1)
```

Call:

```
lm(formula = pressure ~ x + y, data = wolfcamp)
```

Residuals:

| Min      | 1Q      | Median | 3Q     | Max     |
|----------|---------|--------|--------|---------|
| -111.989 | -50.297 | -9.326 | 48.510 | 197.986 |

Coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t )   |
|-------------|-----------|------------|---------|------------|
| (Intercept) | 607.77066 | 7.52219    | 80.80   | <2e-16 *** |
| x           | -1.27844  | 0.06552    | -19.51  | <2e-16 *** |
| y           | -1.13874  | 0.07739    | -14.71  | <2e-16 *** |

---

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

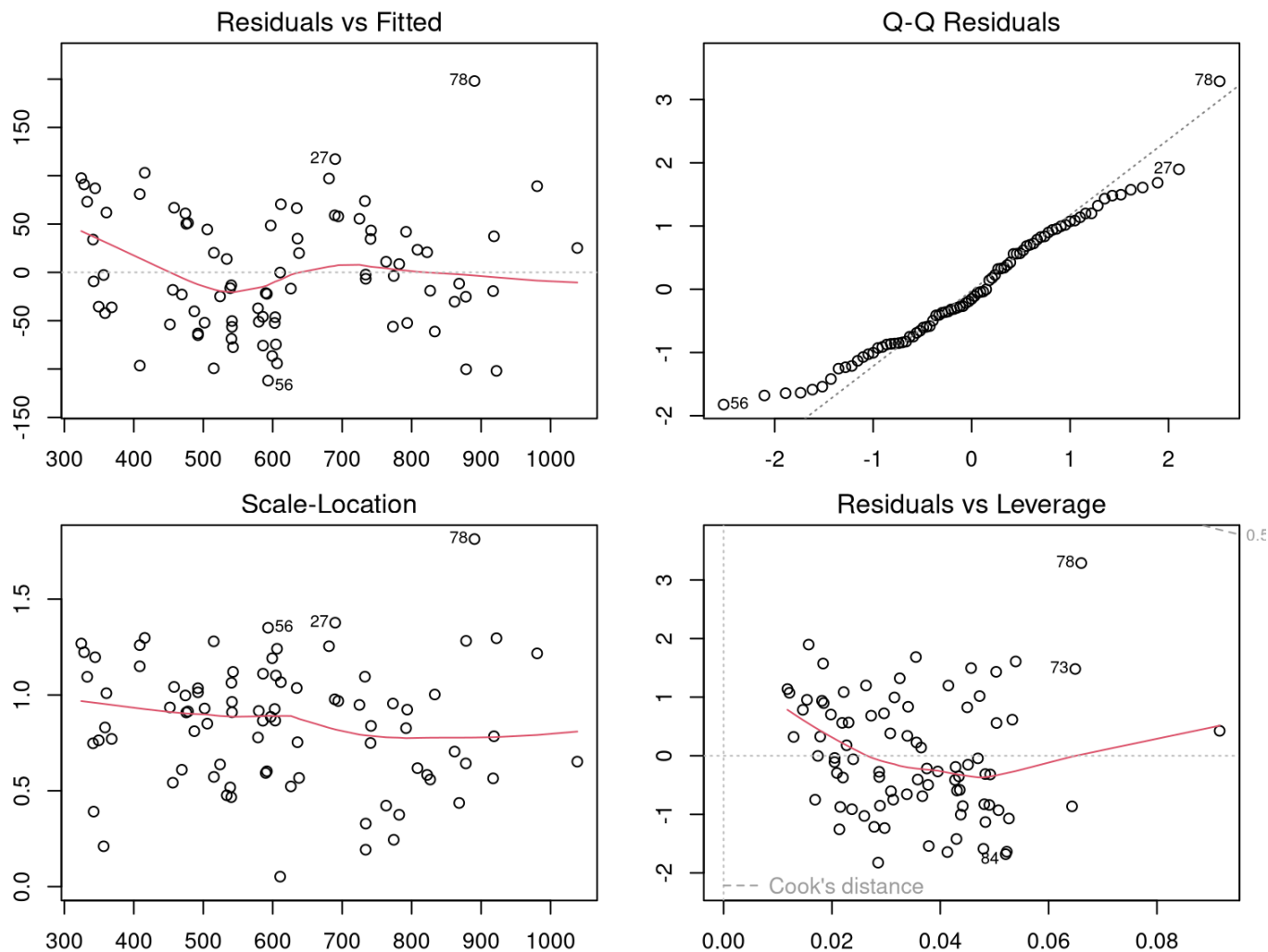
Residual standard error: 62.29 on 82 degrees of freedom

Multiple R-squared: 0.8909, Adjusted R-squared: 0.8882

F-statistic: 334.8 on 2 and 82 DF, p-value: < 2.2e-16

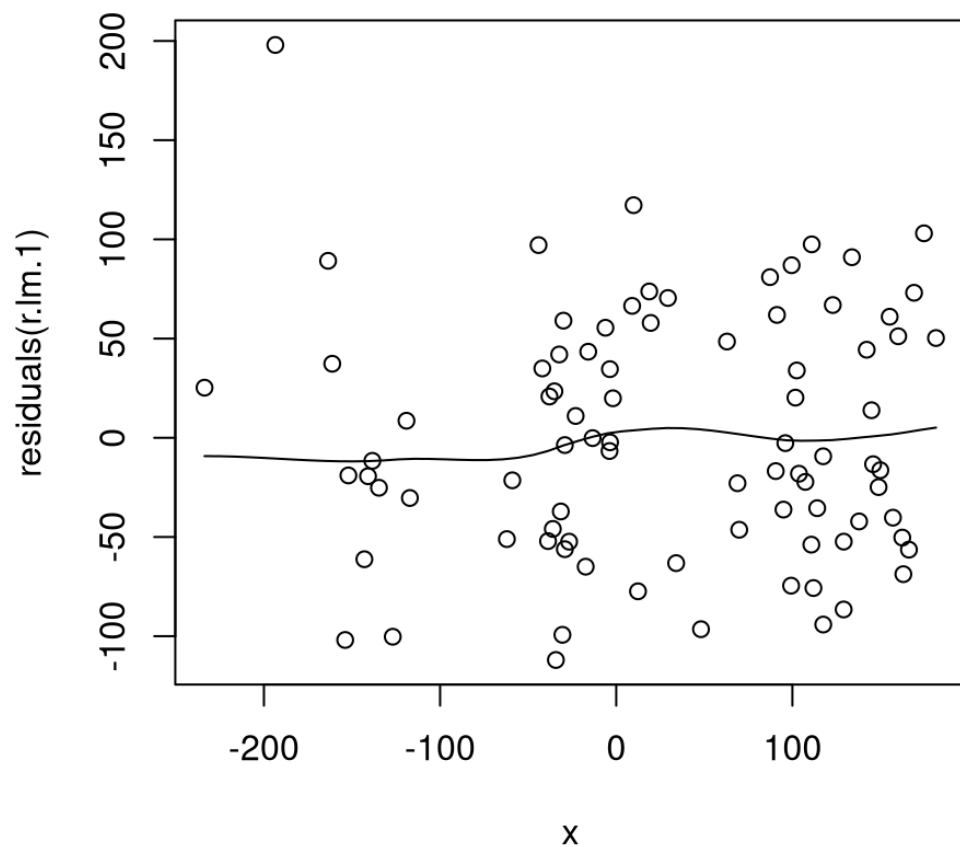
# Residual plots for linear model

```
1 par(mfrow=c(2, 2), mar=c(2,3,2,1))
2 plot(r.lm.1)
```

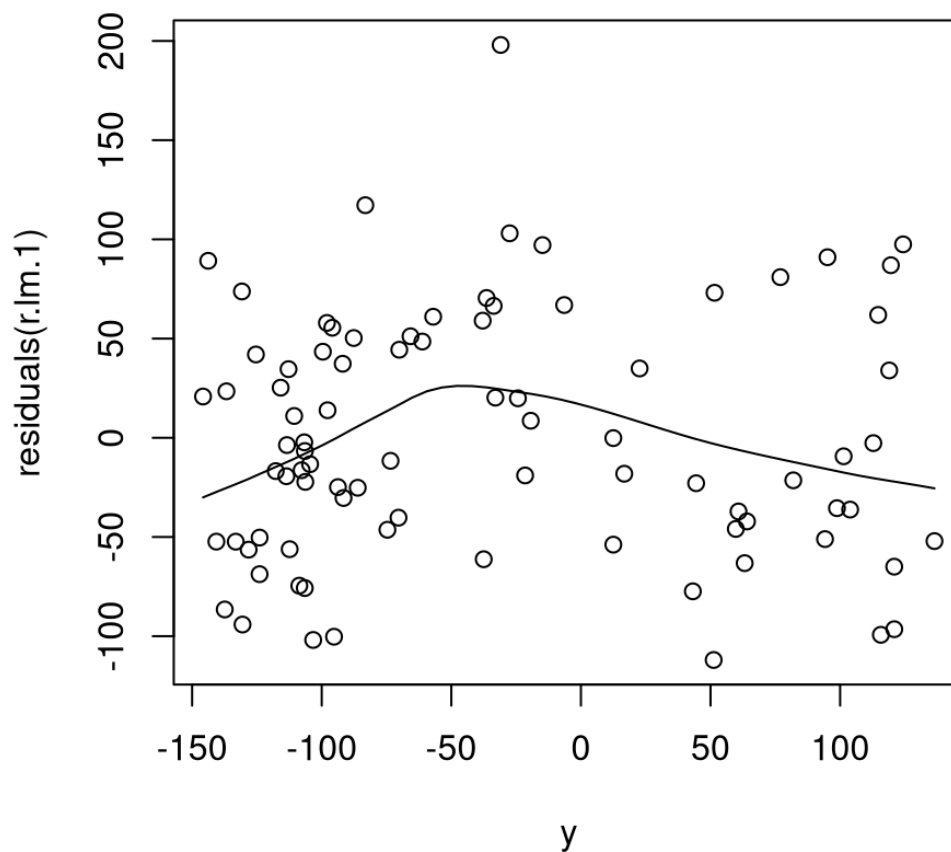


# Residuals plotted against spatial coordinates

residuals vs x



residuals vs y



# Include higher order polynomials

```
1 r.lm.2 <- update(r.lm.1, .~.+I(x^2)+I(y^2)+x:y)
2 summary(r.lm.2)
```

Call:

```
lm(formula = pressure ~ x + y + I(x^2) + I(y^2) + x:y, data = wolfcamp)
```

Residuals:

|  | Min      | 1Q      | Median | 3Q     | Max     |
|--|----------|---------|--------|--------|---------|
|  | -124.405 | -43.662 | -2.337 | 39.017 | 199.198 |

Coefficients:

|             | Estimate   | Std. Error | t value | Pr(> t ) |     |
|-------------|------------|------------|---------|----------|-----|
| (Intercept) | 6.203e+02  | 1.295e+01  | 47.902  | < 2e-16  | *** |
| x           | -1.075e+00 | 8.191e-02  | -13.128 | < 2e-16  | *** |
| y           | -1.330e+00 | 8.861e-02  | -15.008 | < 2e-16  | *** |
| I(x^2)      | 8.994e-05  | 5.908e-04  | 0.152   | 0.879388 |     |
| I(y^2)      | -2.929e-03 | 1.101e-03  | -2.659  | 0.009486 | **  |
| x:y         | 3.184e-03  | 8.790e-04  | 3.622   | 0.000515 | *** |

---

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

Residual standard error: 57.02 on 79 degrees of freedom

Multiple R-squared: 0.9119, Adjusted R-squared: 0.9063

F-statistic: 163.6 on 5 and 79 DF, p-value: < 2.2e-16

# Compare models

```
1 anova(r.lm.1, r.lm.2)
```

Analysis of Variance Table

Model 1: pressure ~ x + y

Model 2: pressure ~ x + y + I(x^2) + I(y^2) + x:y

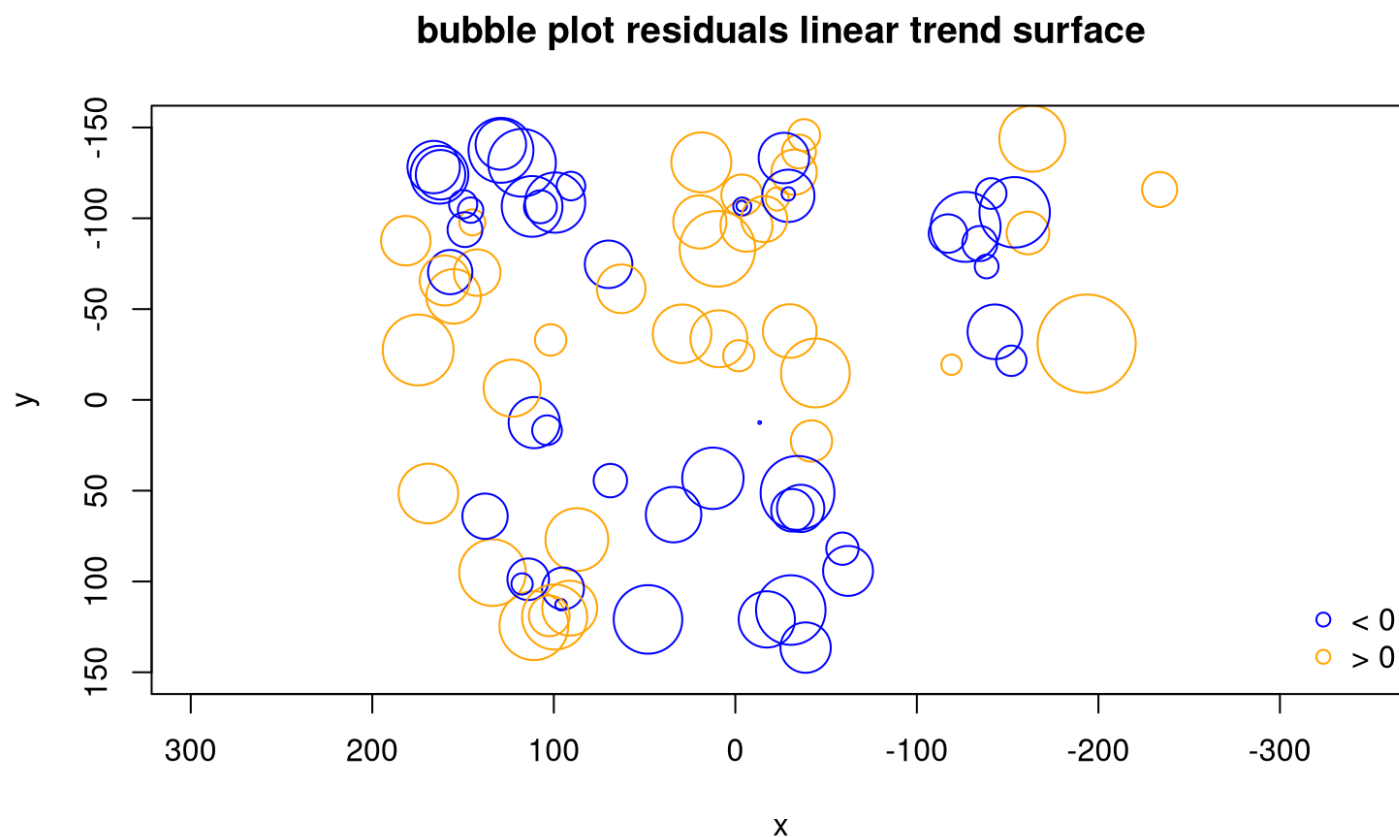
|   | Res.Df | RSS    | Df | Sum of Sq | F      | Pr(>F)       |
|---|--------|--------|----|-----------|--------|--------------|
| 1 | 82     | 318200 |    |           |        |              |
| 2 | 79     | 256887 | 3  | 61313     | 6.2852 | 0.000702 *** |

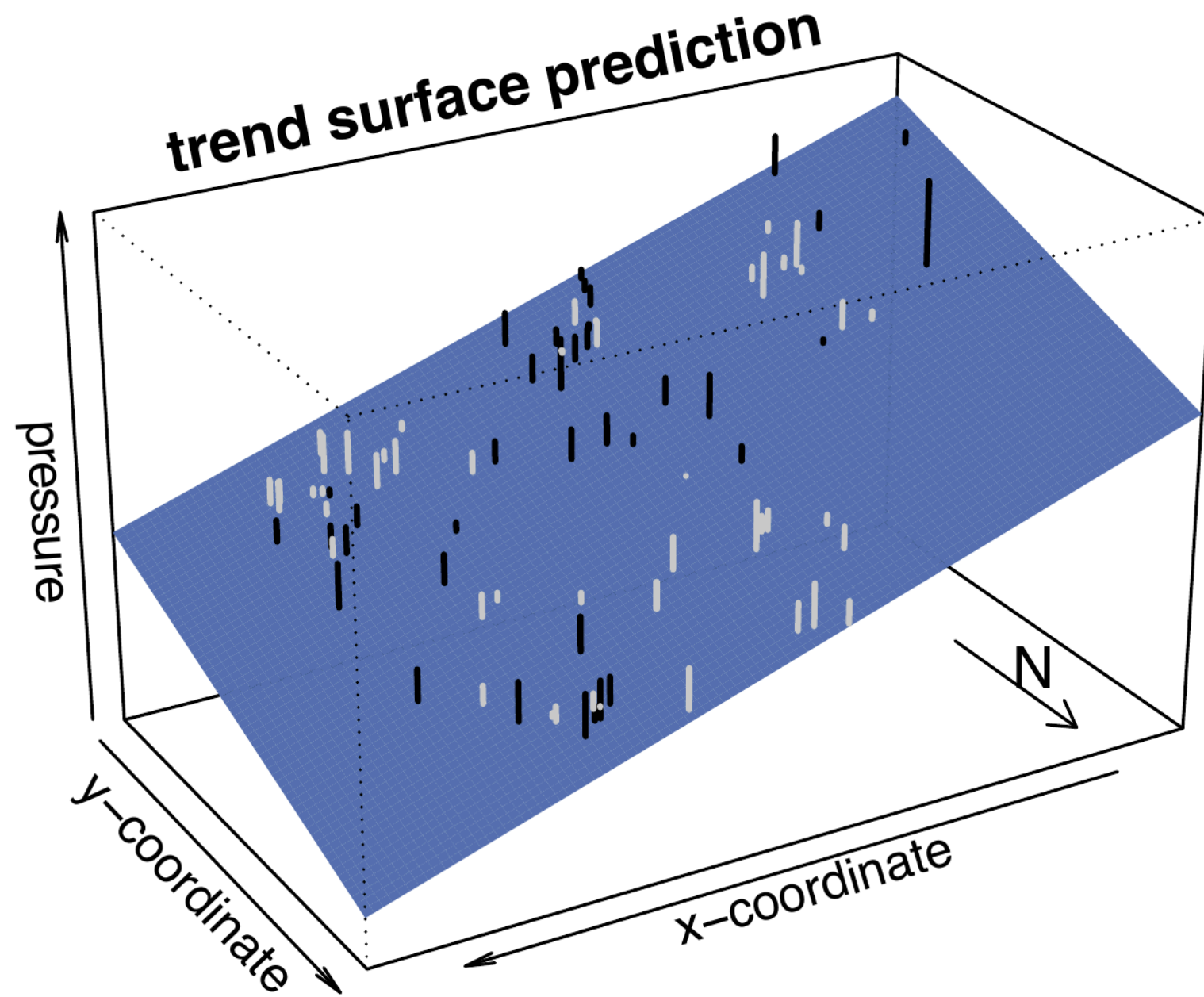
---

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

# Plot residuals of linear trend surface

```
1 plot(y~x, wolfcamp, asp=1, cex=sqrt(abs(residuals(r.lm.1)))/2,  
2      xlim=c(200, -250), ylim=c(150, -150),  
3      col=c("blue", NA, "orange")[sign(residuals(r.lm.1))+2],  
4      main = "bubble plot residuals linear trend surface")  
5 legend("bottomright", pch=1, col=c("blue", "orange"), legend=c("< 0", "> 0"), bty="n")
```





# 5 Modelling auto-correlation

## Pro memoria: sample covariance and correlation

- Data: measurements  $(y_{1,i}, y_{2,i})$ ,  $i = 1, 2, \dots, n$  about 2 response variables
- Sample covariance, where  $\bar{y}_1$  and  $\bar{y}_2$  are the (arithmetic) sample means

$$s_{1,2} = \frac{1}{(n-1)} \sum_{i=1}^n (y_{1,i} - \bar{y}_1)(y_{2,i} - \bar{y}_2)$$

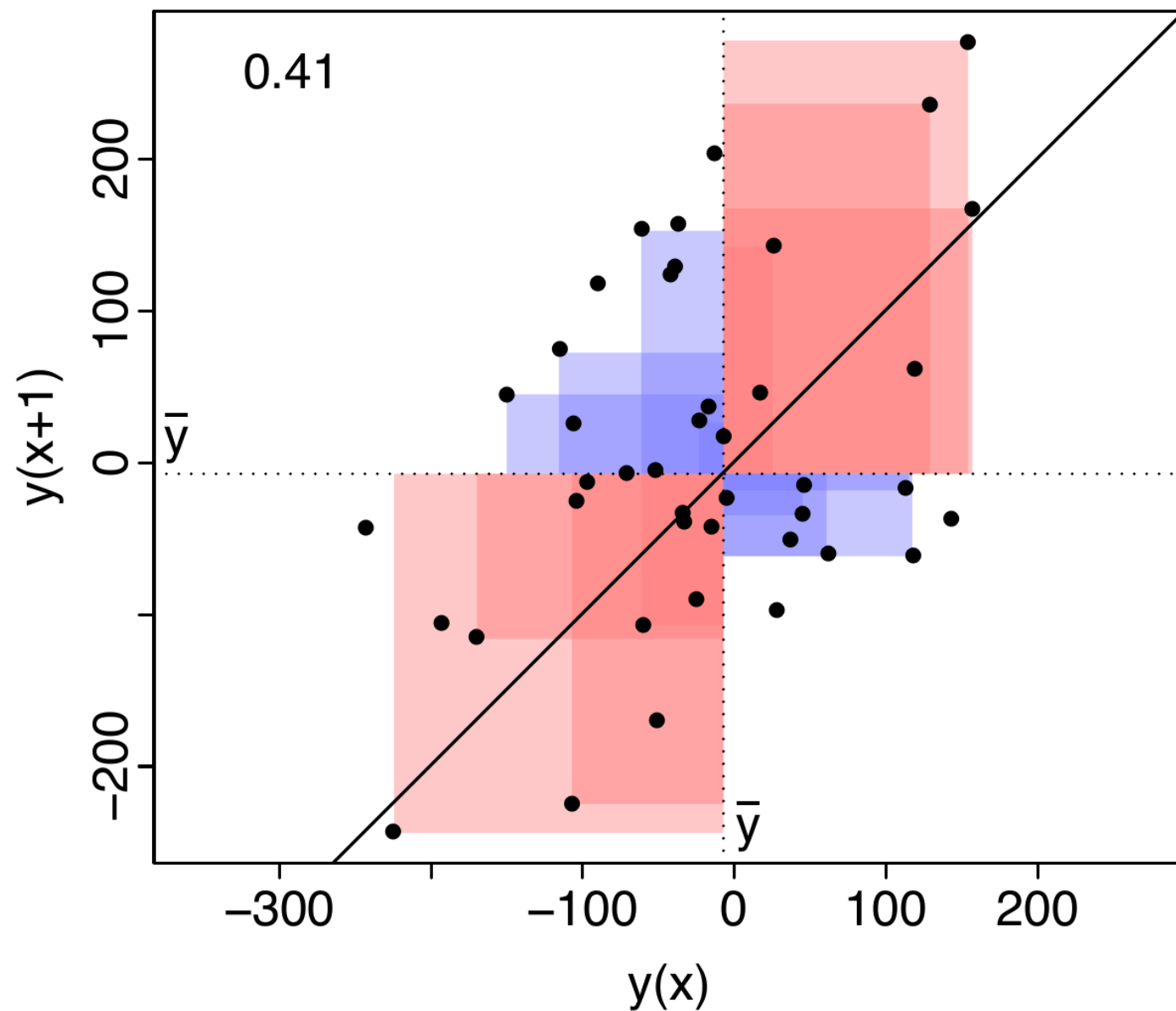
- (Pearson) correlation coefficient, where  $s_1$  and  $s_2$  are the sample standard deviations

$$\hat{\rho} = \frac{s_{1,2}}{s_1 s_2}$$

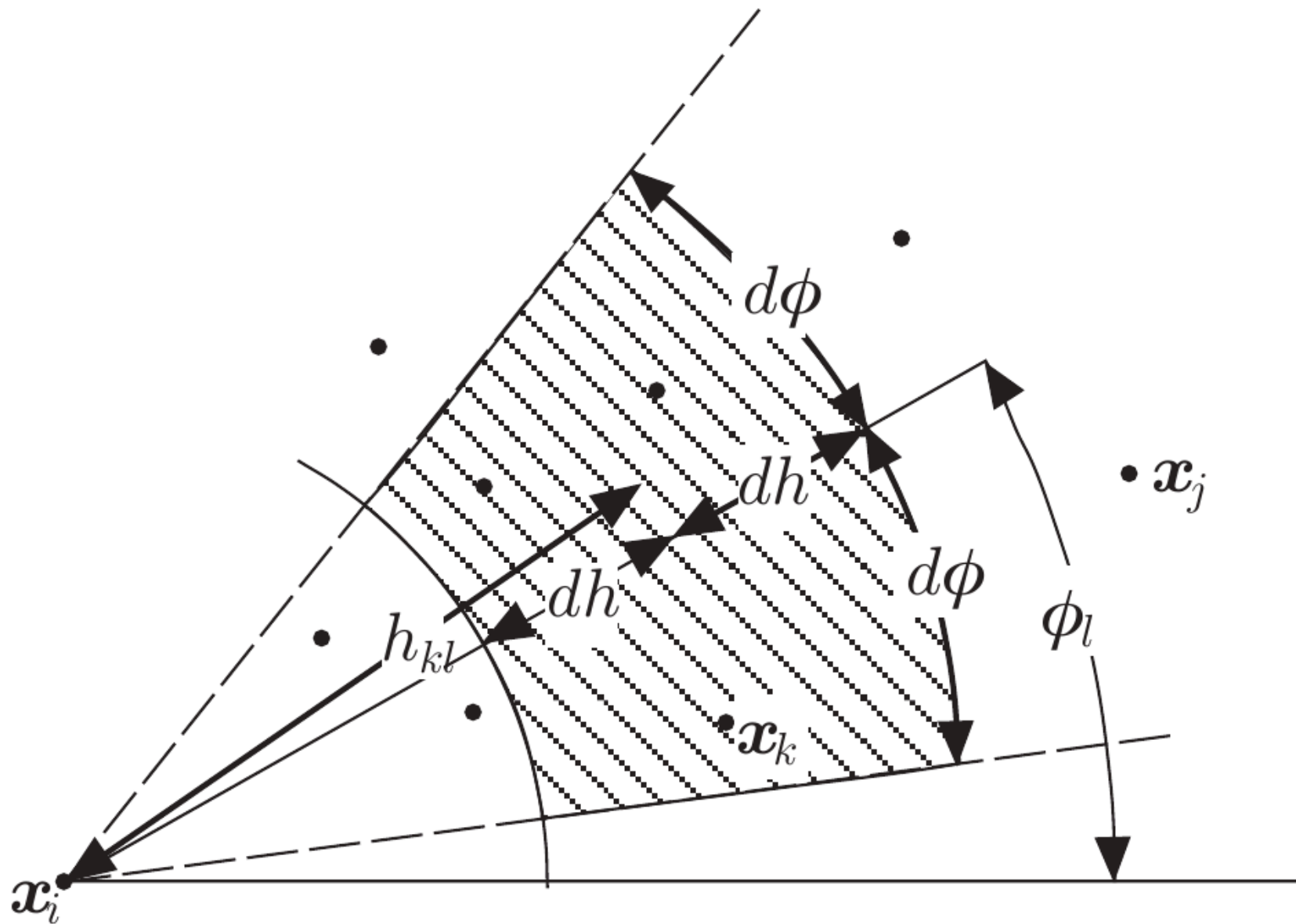
- “plug-in” estimator for auto-correlogram of time series

$$\hat{\rho}(h) = \frac{\sum_{i=1}^{n-h} (y_{i+h} - \bar{y})(y_i - \bar{y})}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

# Covariance and correlation: visual display



# Defining lags for irregular sampling grids



# Spatial Auto-correlation: (co-)variogram of spatial data

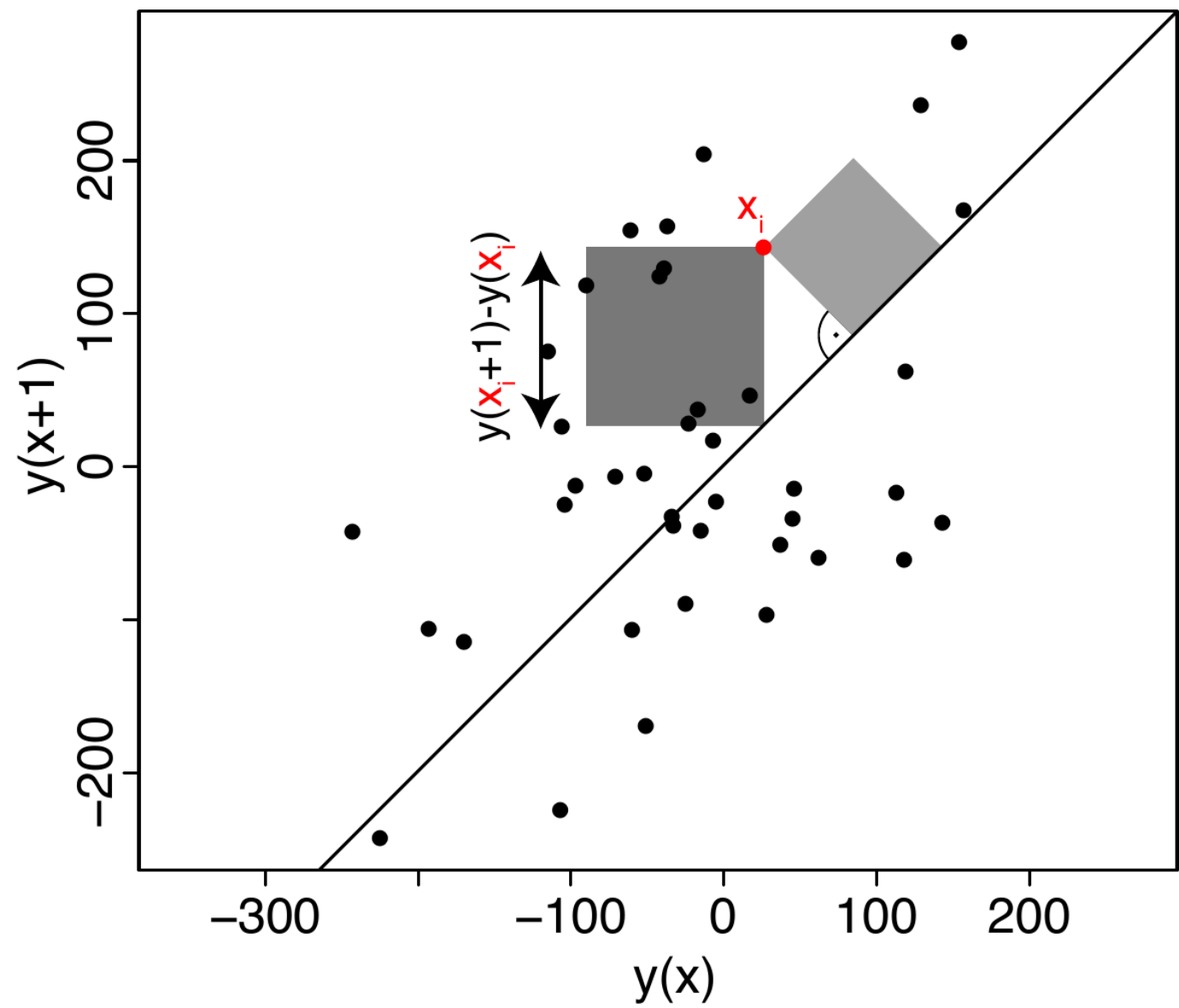
- $(k, l)^{\text{th}}$  lag class,  $\mathbf{h}_{kl}$ , characterized by distance,  $(h_k - dh, h_k + dh]$ , and angular class,  $(\phi_l - d\phi, \phi_l + d\phi]$
- $N_{kl}$ : number of pairs of locations  $(\mathbf{x}_i, \mathbf{x}_j)$  with  $\mathbf{x}_j - \mathbf{x}_i \approx \mathbf{h}_{kl}$
- **Covariogram**: Estimator for covariance for lag class  $\mathbf{h}_{kl}$ :

$$\hat{\gamma}(\mathbf{h}_{kl}) = \frac{1}{N_{kl}} \sum_{(i,j) \in \mathbf{h}_{kl}} [y(\mathbf{x}_i) - \bar{y}][y(\mathbf{x}_j) - \bar{y}]$$

- **(Semi-)variogram**: Estimator for (semi-)variance for lag class  $\mathbf{h}_{kl}$ :

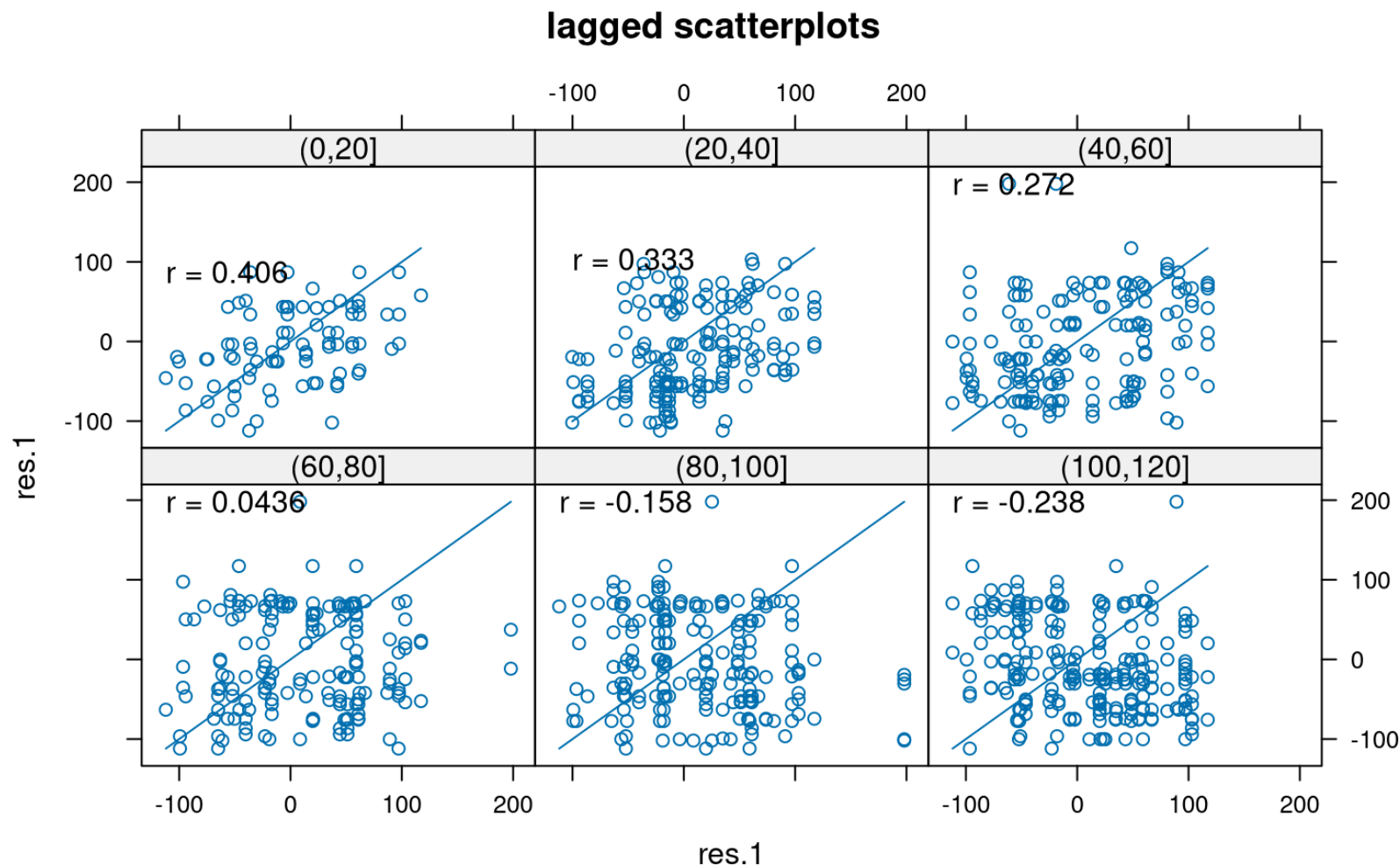
$$\hat{V}(\mathbf{h}_{kl}) = \frac{1}{2 N_{kl}} \sum_{(i,j) \in \mathbf{h}_{kl}} [y(\mathbf{x}_i) - y(\mathbf{x}_j)]^2$$

# Spatial Auto-correlation: Semi-variance



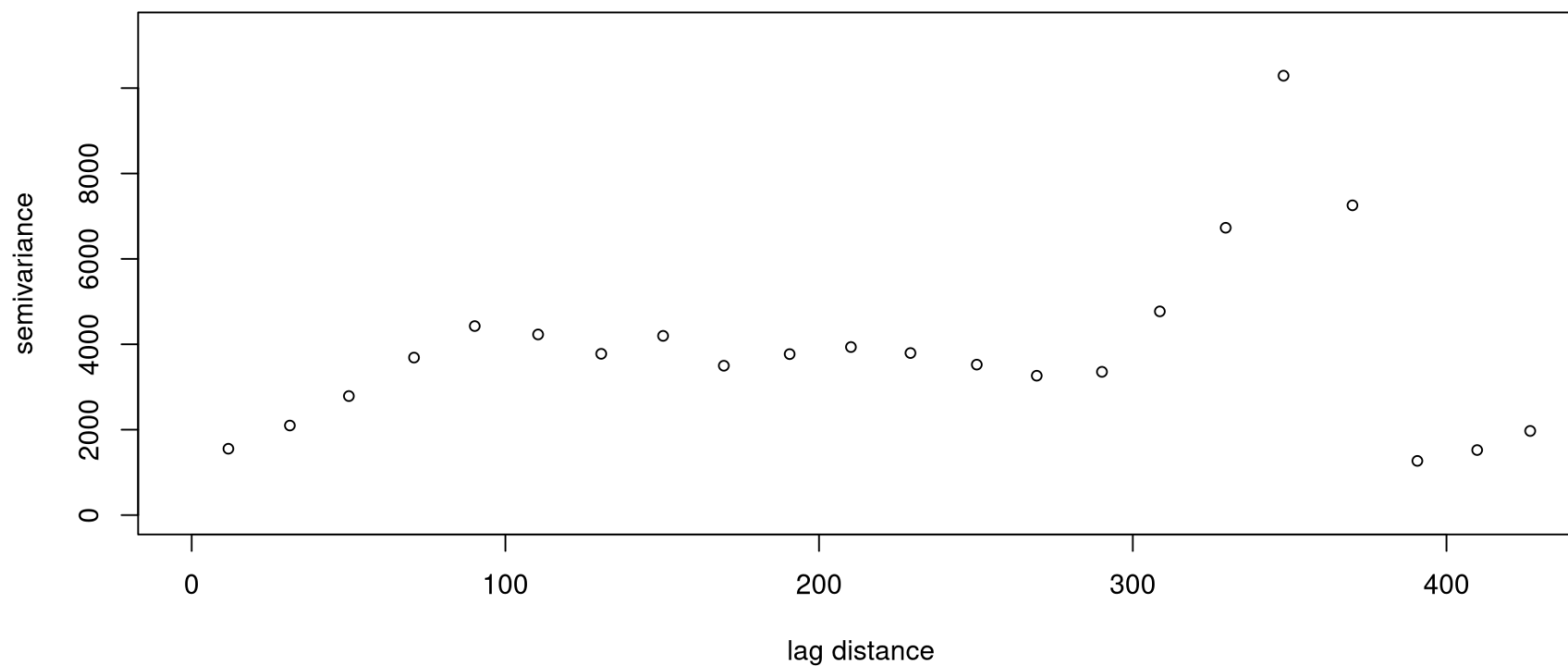
# Lag-scatter plots of trend surface residuals

```
1 d.w$res.1 <- residuals(r.lm.1)
2 hscat(res.1~1, d.w, breaks=seq(0, 120, by=20))
```



# Variogram of trend surface residuals

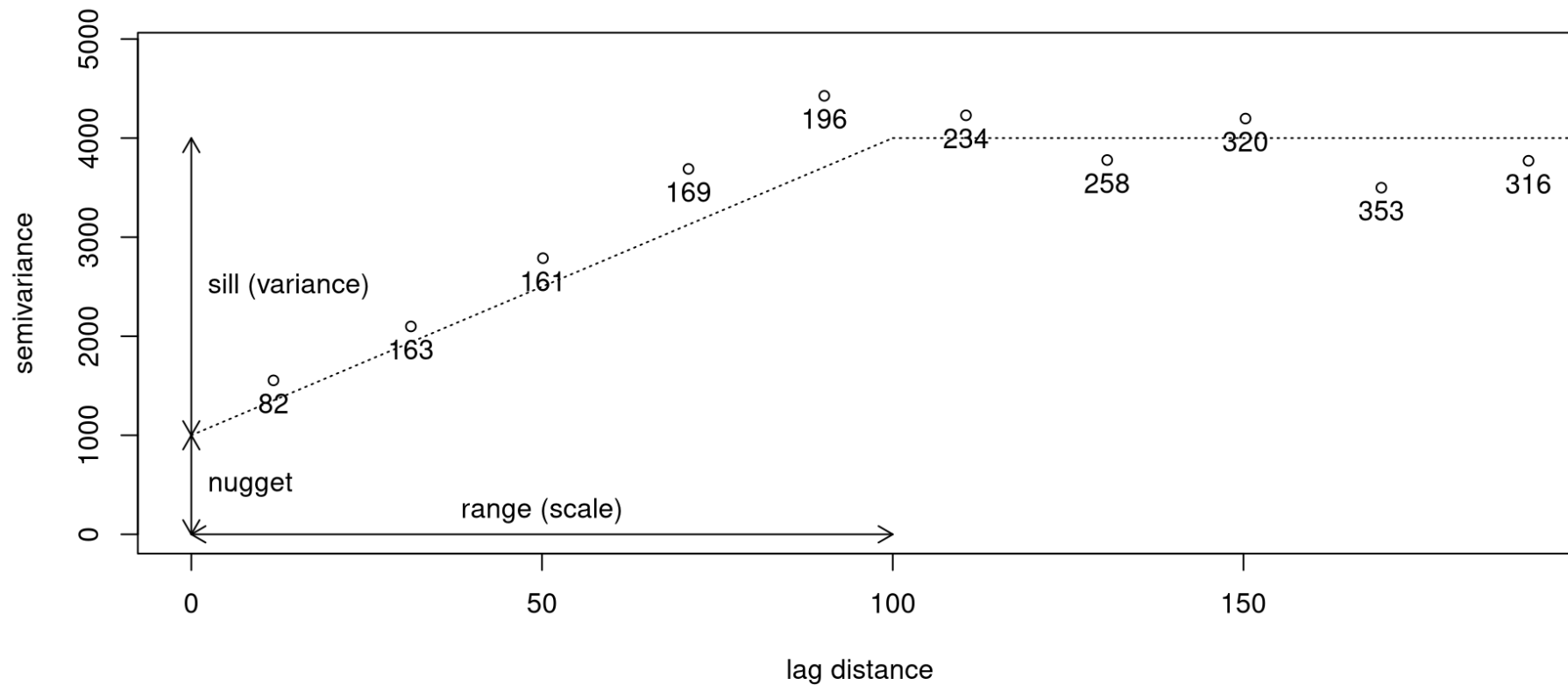
```
1 plot(  
2   sample.variogram(residuals(r.lm.1),  
3   locations=st_coordinates(d.w),  
4   lag.dist.def=20,  
5   estimator="matheron")  
6 )
```



```

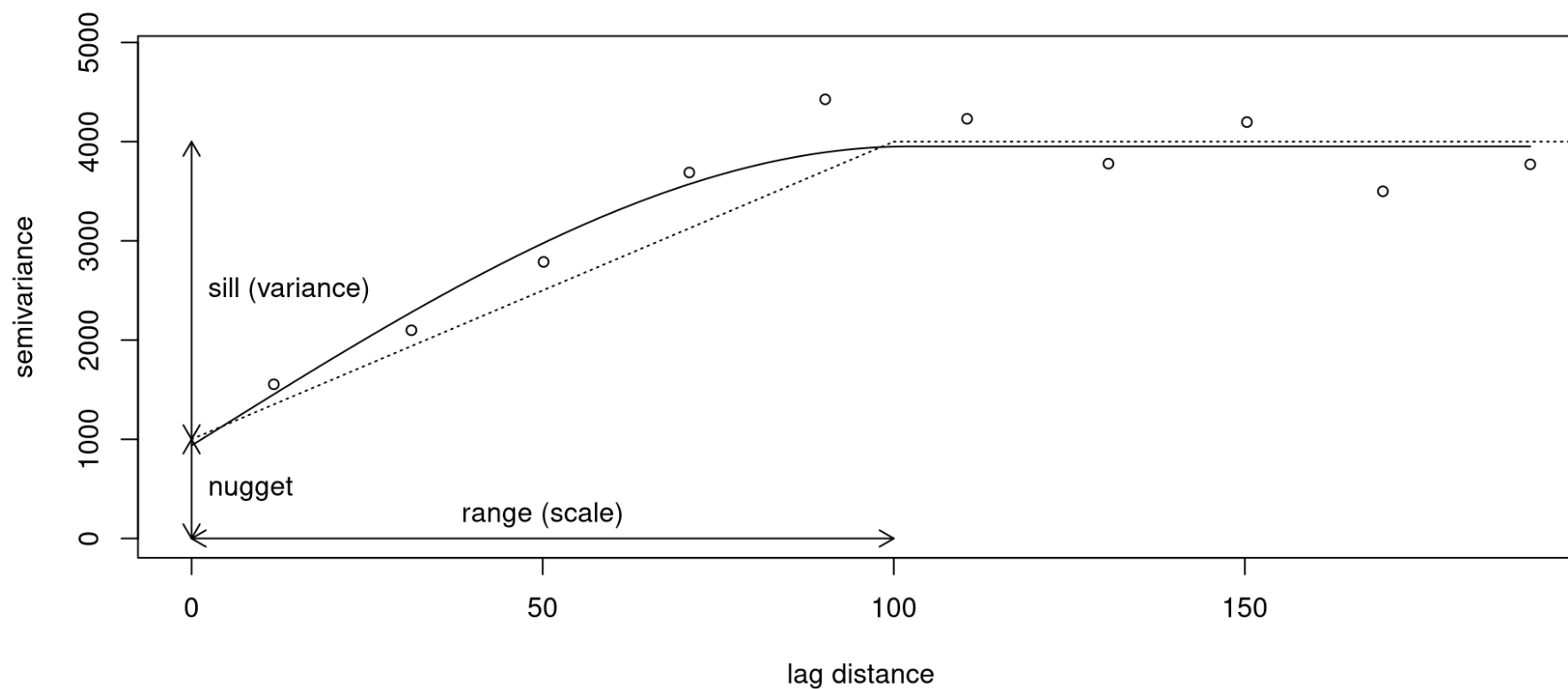
1 r.v <- sample.variogram(residuals(r.lm.1),
2                           locations=st_coordinates(d.w),
3                           lag.dist.def=20,
4                           max.lag=200,
5                           estimator="matheron")
6 plot(r.v)
7 text(gamma~lag.dist, r.v, labels=npairs, pos=1)

```



# Fit variogram function

```
1 r.v.sph <- fit.variogram.model(r.v,  
2                               variogram.model="RMspheric",  
3                               param=c(variance=3000, nugget=1000, scale=100))  
4 plot(r.v)  
5 lines(r.v.sph)
```



# 6 Fitting the spatial model

## Fitting the spatial model with maximum likelihood estimation

```
1 r.georob.1 <- georob(pressure~x+y,  
2                     wolfcamp,  
3                     locations=~x+y,  
4                     variogram.model="RMspheric",  
5                     param=c(variance=3000, nugget=1000, scale=100),  
6                     tuning.psi=1000,  
7                     control=control.georob(ml.method="ML"))  
8 summary(r.georob.1)
```

```
...  
Maximized log-likelihood: -458.3671
```

```
...  
Variogram: RMspheric
```

|                | Estimate | Lower   | Upper  |
|----------------|----------|---------|--------|
| variance       | 3328.90  | 1453.95 | 7621.7 |
| snugget(fixed) | 0.00     | NA      | NA     |
| nugget         | 1236.27  | 616.01  | 2481.1 |
| scale          | 122.95   | 95.13   | 158.9  |

```
Fixed effects coefficients:
```

|             | Estimate | Std. Error | t value | Pr(> t ) |
|-------------|----------|------------|---------|----------|
| (Intercept) | 620.3550 | 17.0641    | 36.354  | < 2e-16  |
| x           | -1.3256  | 0.1360     | -9.750  | 2.33e-15 |
| y           | -1.2061  | 0.1793     | -6.727  | 2.16e-09 |

```
...
```

# Compare to linear model

```
1 summary(r.lm.1)
```

Call:

```
lm(formula = pressure ~ x + y, data = wolfcamp)
```

Residuals:

| Min      | 1Q      | Median | 3Q     | Max     |
|----------|---------|--------|--------|---------|
| -111.989 | -50.297 | -9.326 | 48.510 | 197.986 |

Coefficients:

|             | Estimate  | Std. Error | t value | Pr(> t )   |
|-------------|-----------|------------|---------|------------|
| (Intercept) | 607.77066 | 7.52219    | 80.80   | <2e-16 *** |
| x           | -1.27844  | 0.06552    | -19.51  | <2e-16 *** |
| y           | -1.13874  | 0.07739    | -14.71  | <2e-16 *** |

---

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

Residual standard error: 62.29 on 82 degrees of freedom

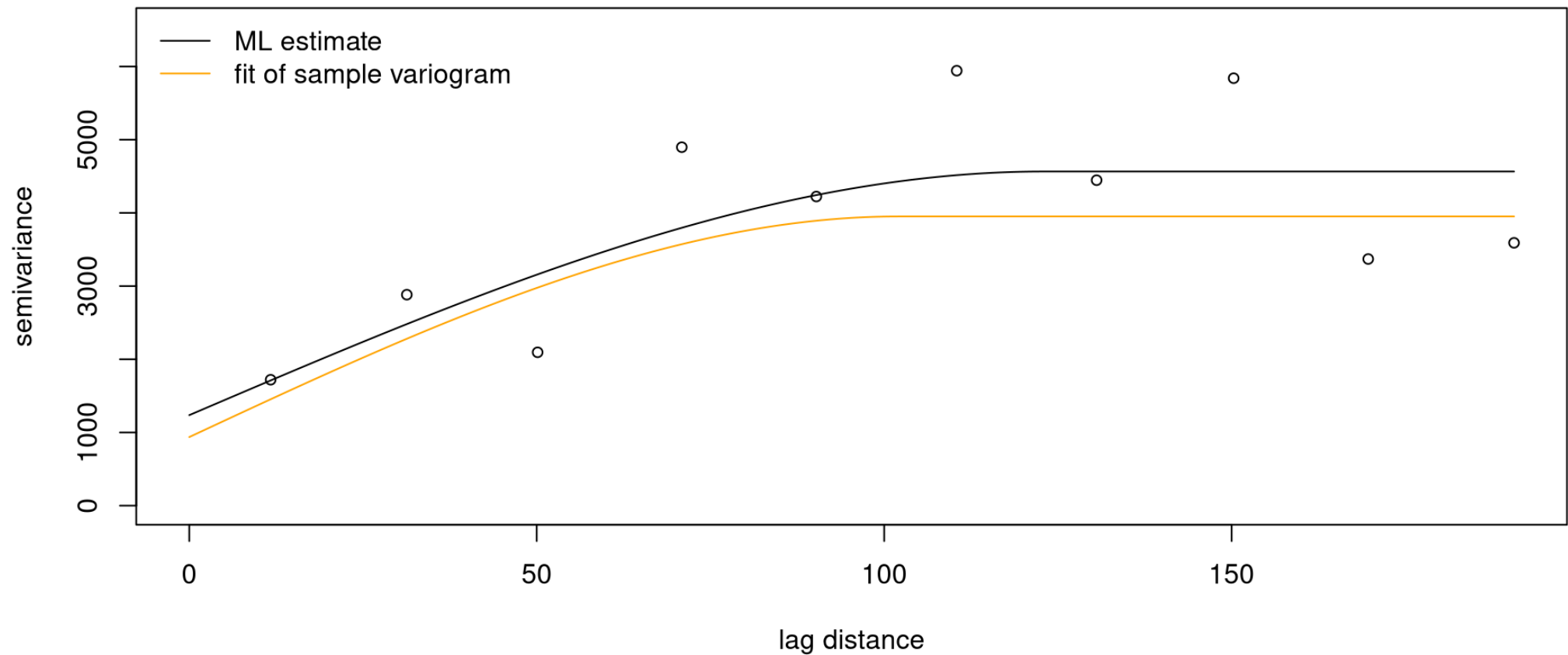
Multiple R-squared: 0.8909, Adjusted R-squared: 0.8882

F-statistic: 334.8 on 2 and 82 DF, p-value: < 2.2e-16

```

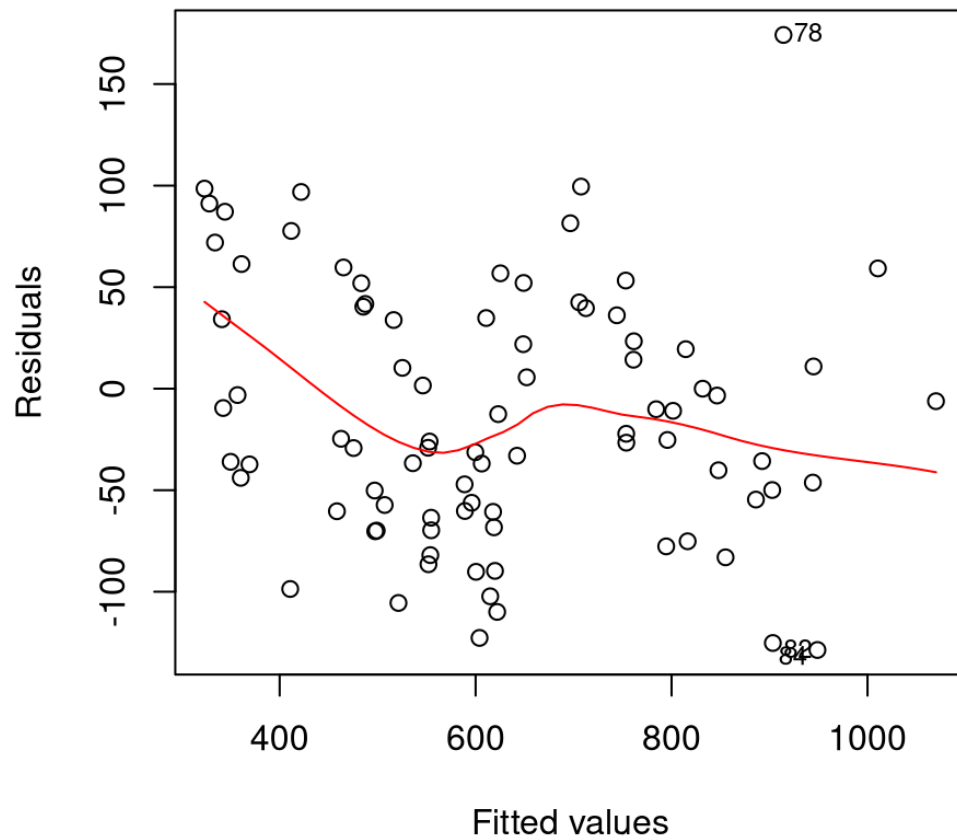
1 plot(r.georob.1, lag.dist.def=20, max.lag=200)
2 lines(r.v.sph, col="orange")
3 legend("topleft", lty=1, col=c("black", "orange"),
4       bty="n", legend=c("ML estimate", "fit of sample variogram"))

```

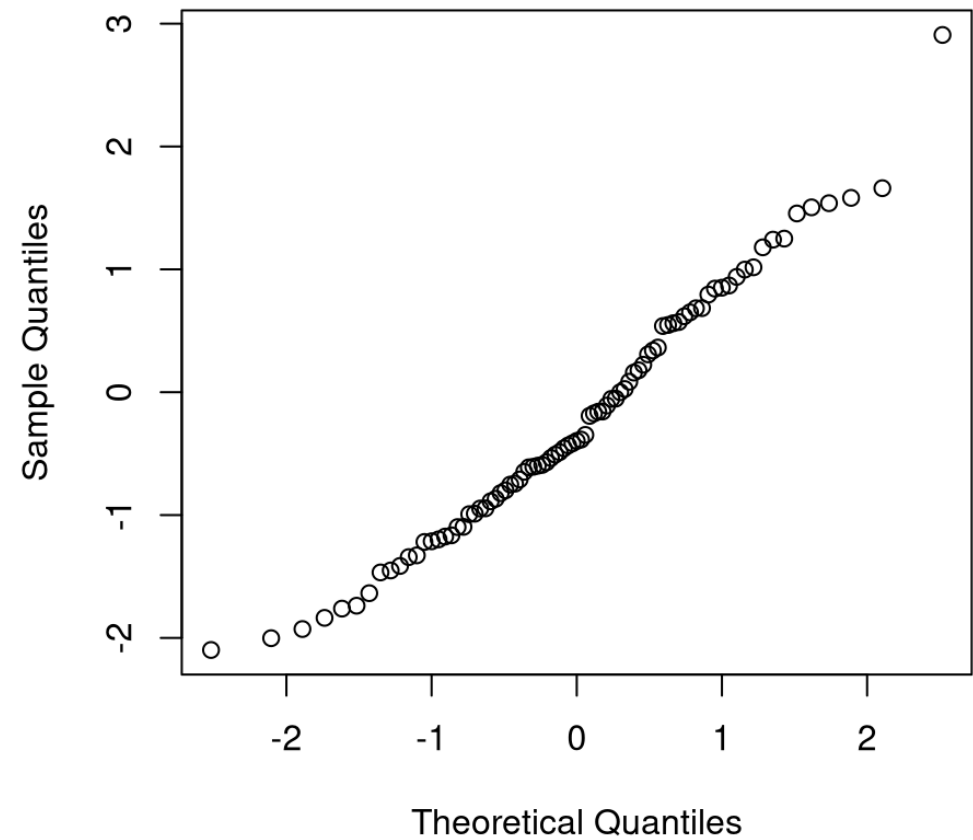


```
1 par(mfrow=c(1,2))
2 plot(r.georob.1, what = "ta")
3 qqnorm(rstandard(r.georob.1, level=0), main="QQnorm regression residuals")
```

**Residuals vs. Fitted**



**QQnorm regression residuals**



# 7 Model building and assessment

Data analysis often leads to a set of equally plausible candidate models that use different set of covariates and different variograms

- compare fit of candidate models by hypothesis tests taking auto-correlation properly into account
- use established goodness-of-fit criteria (AIC, BIC) to select a “best” model, again taking auto-correlation into account
- use cross-validation to compare the power of candidate models to *predict new data*

# ML fit quadratic trend surface model

```
1 r.georob.2 <- update(r.georob.1, .~.+I(x^2)+I(y^2)+x:y)
2 summary(r.georob.2)
```

```
...
Maximized log-likelihood: -455.0776
```

```
...
Variogram: RMspheric
      Estimate  Lower  Upper
variance    2060.36  784.17 5413.4
snugget(fixed)    0.00    NA    NA
nugget        1402.08  654.37 3004.1
scale          103.85   30.07  358.7
```

Fixed effects coefficients:

|             | Estimate   | Std. Error | t value | Pr(> t ) |
|-------------|------------|------------|---------|----------|
| (Intercept) | 6.111e+02  | 2.165e+01  | 28.225  | < 2e-16  |
| x           | -1.168e+00 | 1.317e-01  | -8.868  | 1.76e-13 |
| y           | -1.268e+00 | 1.566e-01  | -8.098  | 5.61e-12 |
| I(x^2)      | 1.297e-03  | 9.419e-04  | 1.377   | 0.172    |
| I(y^2)      | -2.319e-03 | 1.652e-03  | -1.404  | 0.164    |
| x:y         | 2.290e-03  | 1.500e-03  | 1.526   | 0.131    |

```
1 waldtest(r.georob.2, r.georob.1, test="F")
```

Wald test

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

Model 2: pressure ~ x + y

|   | Res.Df | Df | F      | Pr(>F)  |   |
|---|--------|----|--------|---------|---|
| 1 | 79     |    |        |         |   |
| 2 | 82     | -3 | 2.7684 | 0.04713 | * |

---

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

```
1 step(r.georob.2)
```

Start: AIC=922.16

pressure ~ x + y + I(x^2) + I(y^2) + x:y

|          | Df | AIC    | Converged |
|----------|----|--------|-----------|
| - I(x^2) | 1  | 922.05 | 1         |
| - I(y^2) | 1  | 922.13 | 1         |
| <none>   |    | 922.16 |           |
| - x:y    | 1  | 922.49 | 1         |

Step: AIC=922.05

pressure ~ x + y + I(y^2) + x:y

|          | Df | AIC    | Converged |
|----------|----|--------|-----------|
| <none>   |    | 922.05 |           |
| + I(x^2) | 1  | 922.16 | 1         |
| - I(y^2) | 1  | 922.54 | 1         |
| - x:y    | 1  | 924.61 | 1         |

Tuning constant: 1000

Fixed effects coefficients:

| (Intercept) | x         | y         | I(y^2)    | x:y      |
|-------------|-----------|-----------|-----------|----------|
| 627.526464  | -1.148338 | -1.347008 | -0.002587 | 0.003005 |

Variogram: RMspheric

| variance(fixed) | snugget(fixed) | nugget(fixed) | scale(fixed) |
|-----------------|----------------|---------------|--------------|
| 2060.4          | 0.0            | 1402.1        | 103.8        |

# 8 Cross-validating trend surfaces

```
1 r.cv.1 <- cv(r.georob.1, seed=5426, method = "random")
2 r.cv.2 <- cv(r.georob.2, seed=5426, method = "random")
```

```
1 summary(r.cv.1)
```

Statistics of cross-validation prediction errors

| me     | mede    | rmse    | made    | qne     | msse   | medsse | crps    |
|--------|---------|---------|---------|---------|--------|--------|---------|
| 2.6188 | 10.3138 | 53.4124 | 48.8261 | 51.8898 | 1.1472 | 0.5079 | 29.4385 |

```
1 summary(r.cv.2)
```

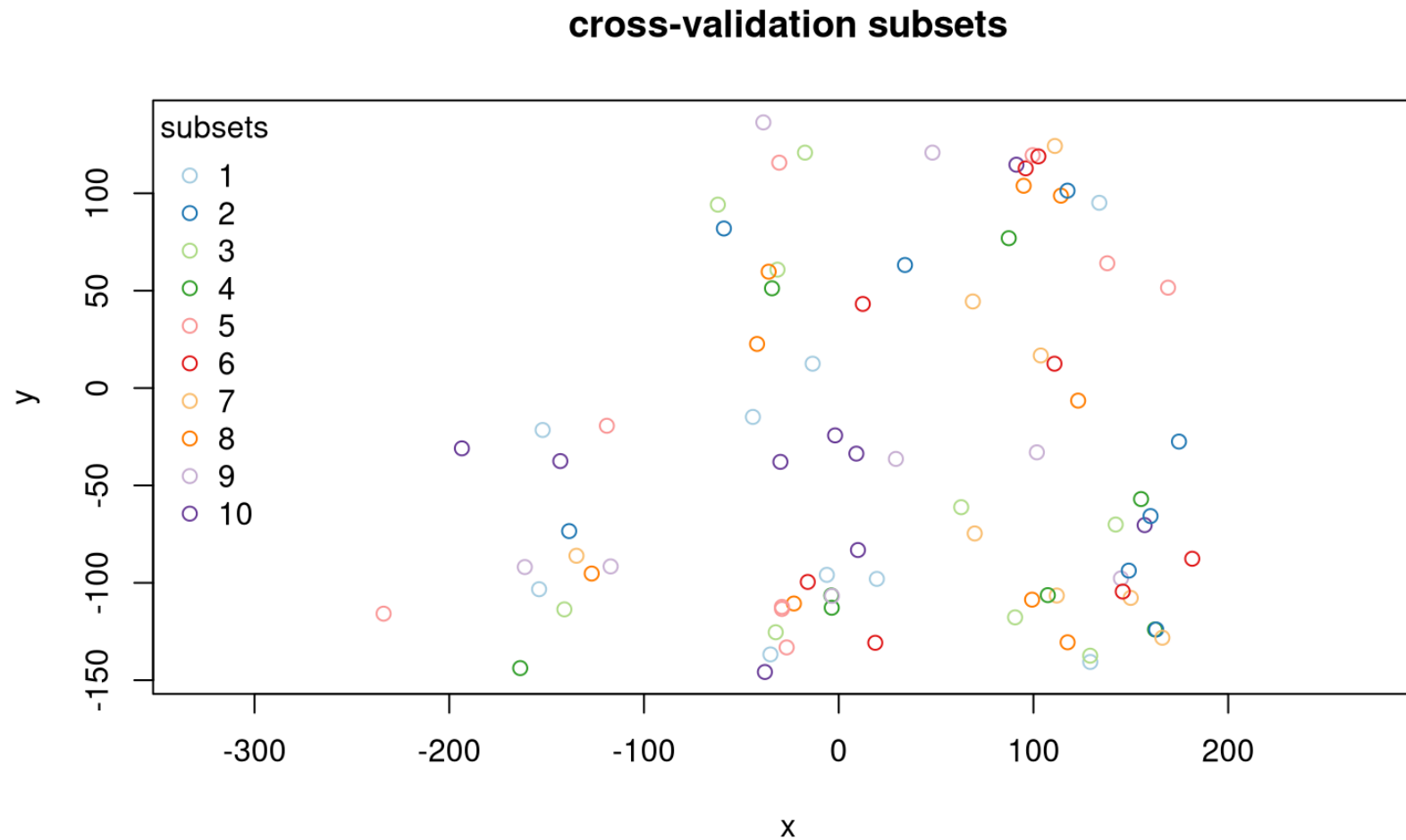
Statistics of cross-validation prediction errors

| me     | mede   | rmse    | made    | qne     | msse   | medsse | crps    |
|--------|--------|---------|---------|---------|--------|--------|---------|
| 0.5079 | 8.6112 | 56.8230 | 50.1898 | 52.4405 | 1.2229 | 0.5493 | 30.9231 |

```

1 library(RColorBrewer)
2 pal <- brewer.pal(10, "Paired")
3 plot(y~x, r.cv.1$pred, asp=1, col= pal[subset], main = "cross-validation subsets")
4 legend("topleft", pch = 1, col = pal, title = "subsets", legend = 1:10, bty="n")

```

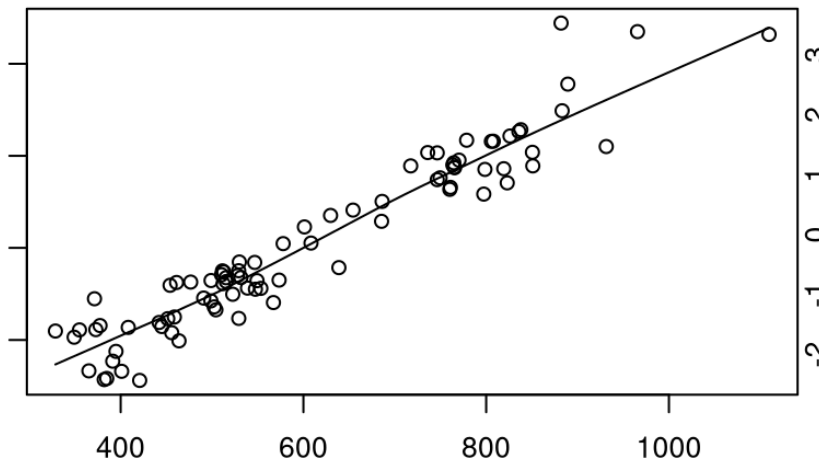


```

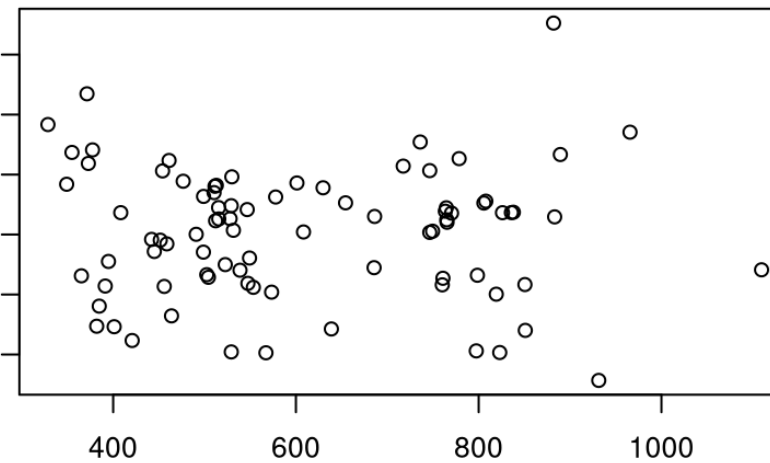
1 par(mfrow=c(2, 2), mar = c(2,1,2,1))
2 plot(r.cv.1, type="sc")
3 plot(r.cv.1, type="ta")
4 plot(r.cv.1, type="qq") #| plot(r.cv.1, type="hist.pit")

```

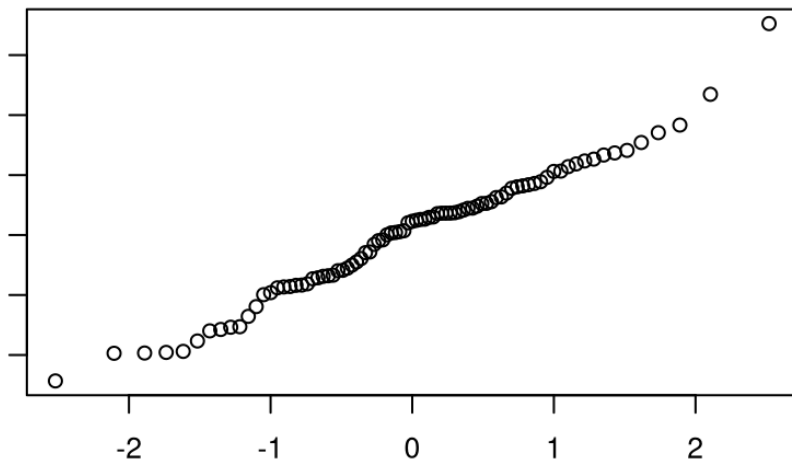
**data vs. predictions**



**Tukey-Anscombe plot**



**normal-QQ-plot of standardized prediction errors**



# 9 Spatial predictions by kriging

- Prediction of signal  $S(\mathbf{x}_0)$  at location  $\mathbf{x}_0$  without measurement

$$\hat{S}(\mathbf{x}_0) = \sum_{i=1}^n \kappa_i(\mathbf{x}_0) y(\mathbf{x}_i)$$

- Where weights  $\kappa_i(\mathbf{x}_0)$  depend on trend model and variogram
- Kriging provides in addition an estimate of the variance of the prediction error  $S(\mathbf{x}_0) - \hat{S}(\mathbf{x}_0)$

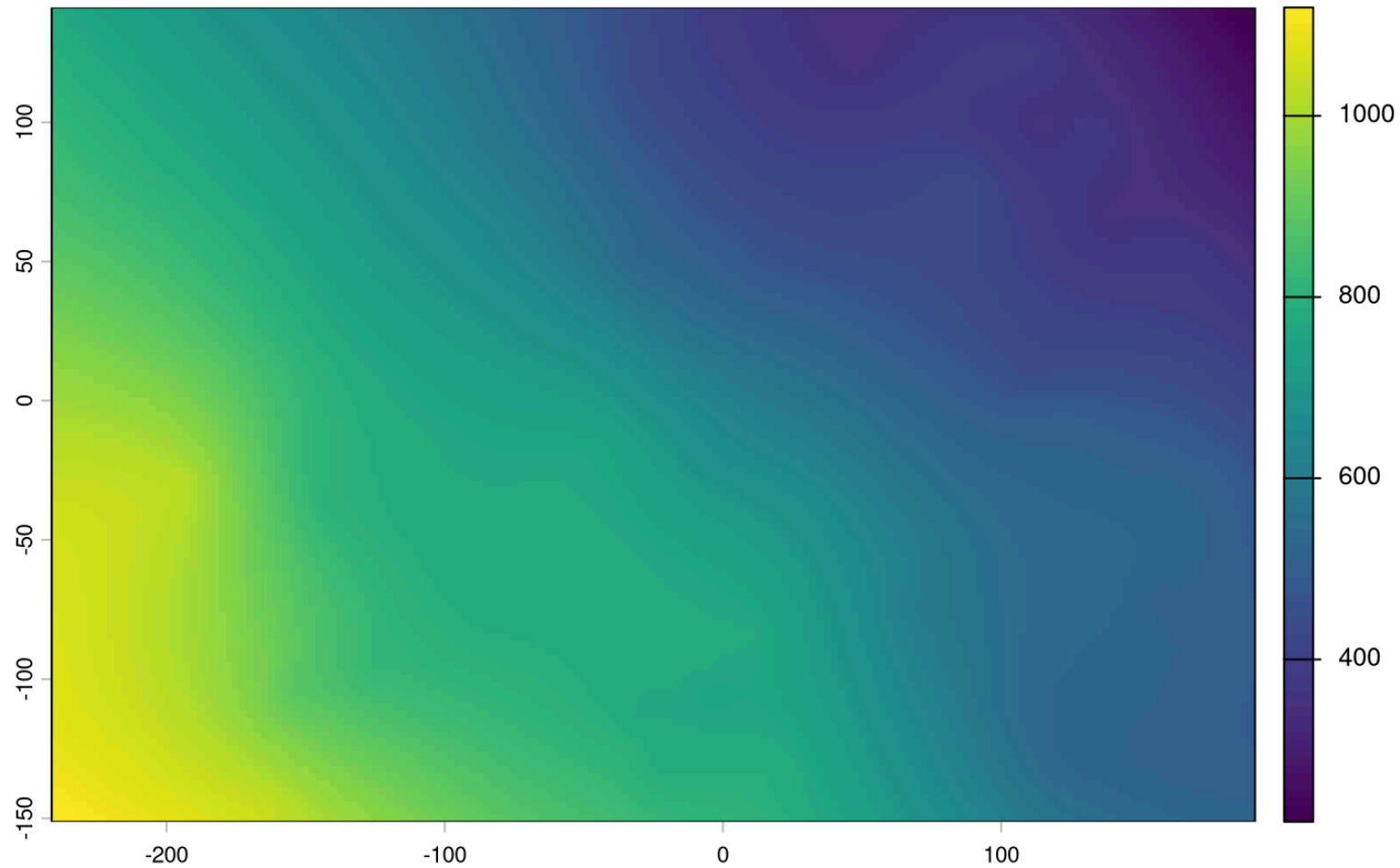
# Compute spatial predictions for Wolfcamp

```
1 d.w.grid <- expand.grid(  
2   x = seq(-240, 190, by= 2.5),  
3   y = seq(-150, 140, by= 2.5)  
4 )  
5 r.uk <- predict(r.georob.1, newdata=d.w.grid)  
6 r.uk <- rast(r.uk)  
7 r.uk
```

```
class      : SpatRaster  
size       : 117, 173, 4  (nrow, ncol, nlyr)  
resolution : 2.5, 2.5  (x, y)  
extent     : -241.25, 191.25, -151.25, 141.25  (xmin, xmax, ymin, ymax)  
coord. ref.:  
source(s) : memory  
names      :      pred,      se,      lower,      upper  
min values : 220.7894, 18.94777,  92.88801, 348.6907  
max values : 1119.7373, 77.22483, 1012.69432, 1226.7803
```

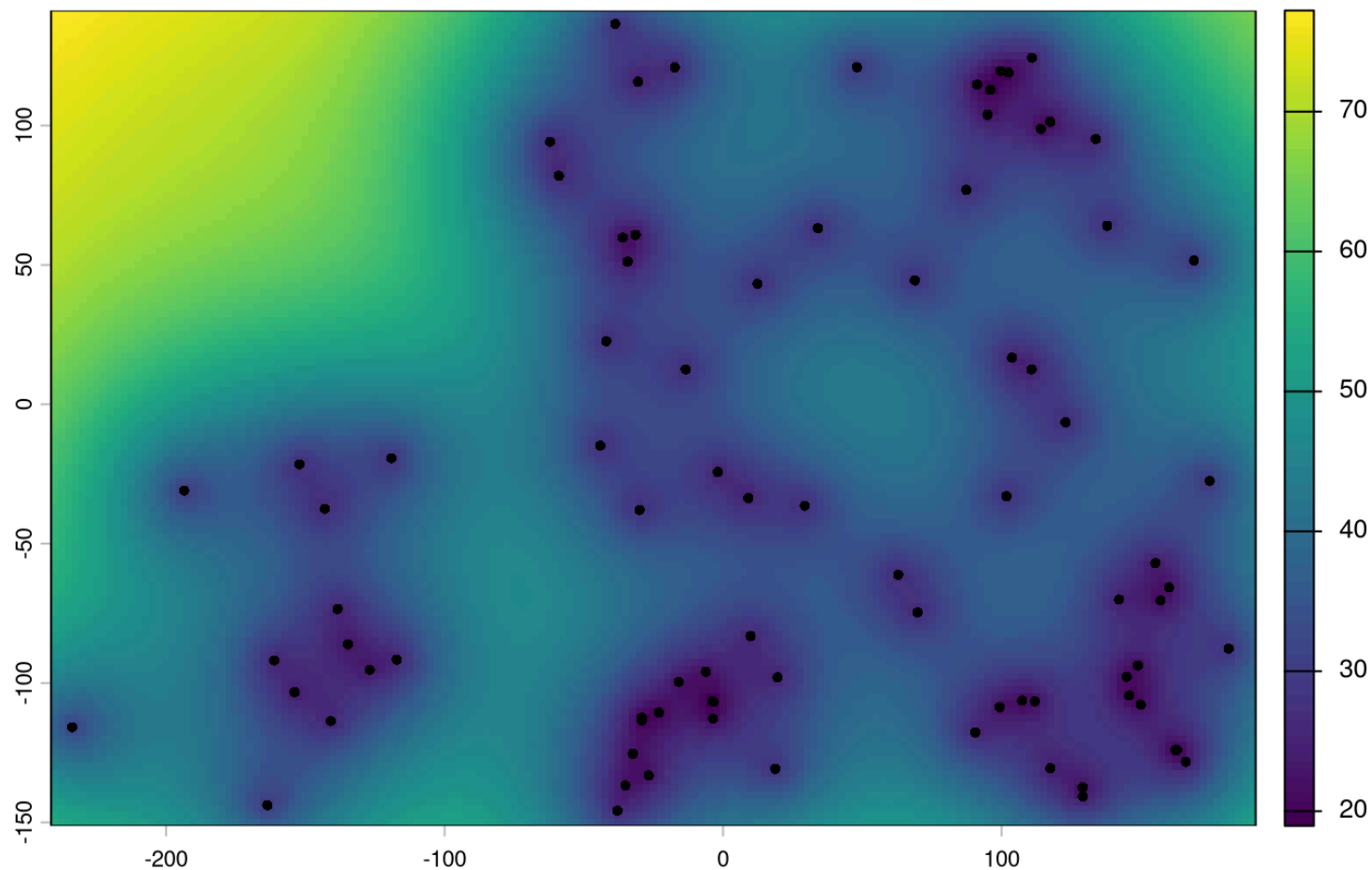
# Universal kriging predictions

```
1 plot(r.uk["pred"])
```



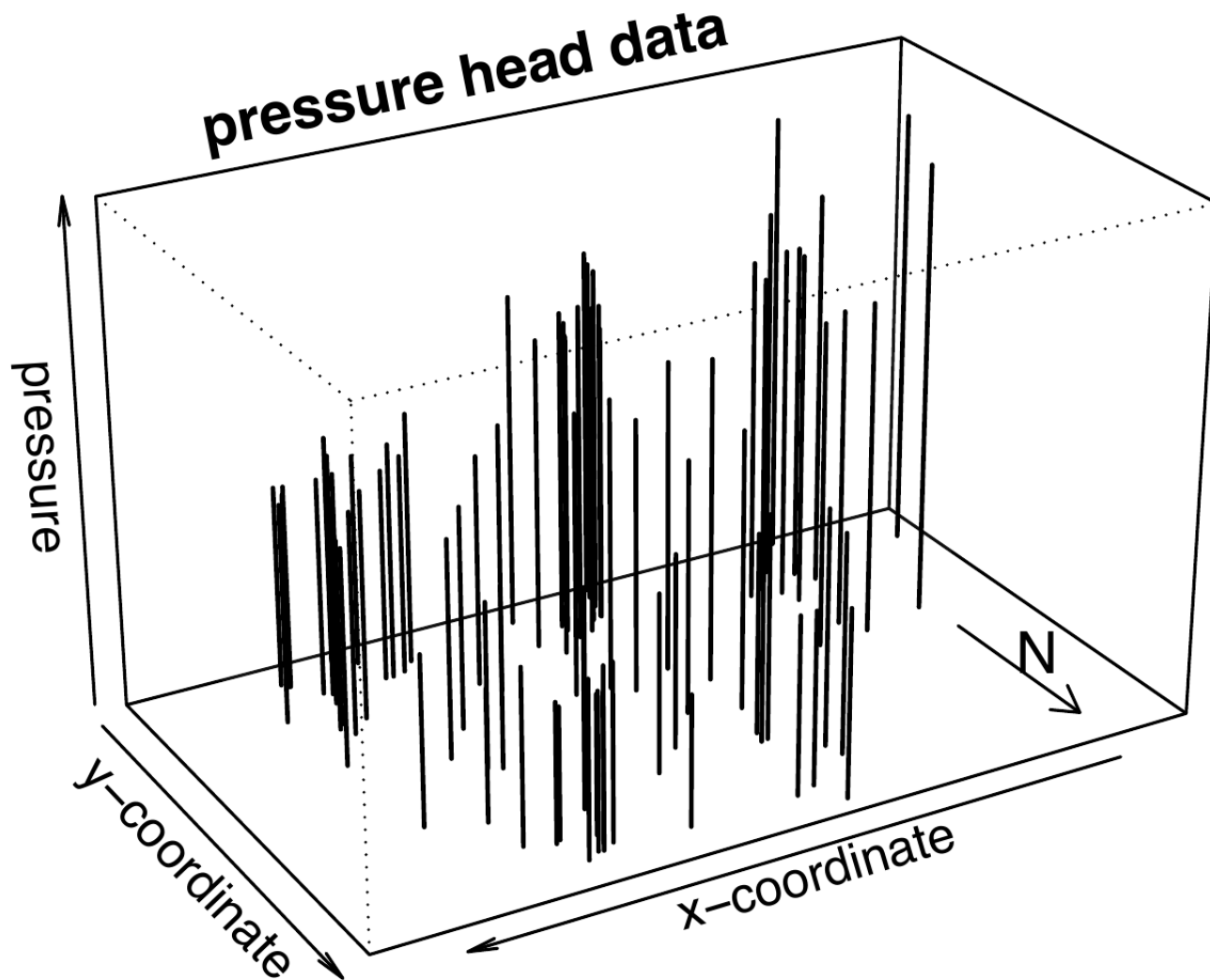
# Universal kriging standard errors

```
1 plot(r.uk["se"])
2 points(d.w)
```

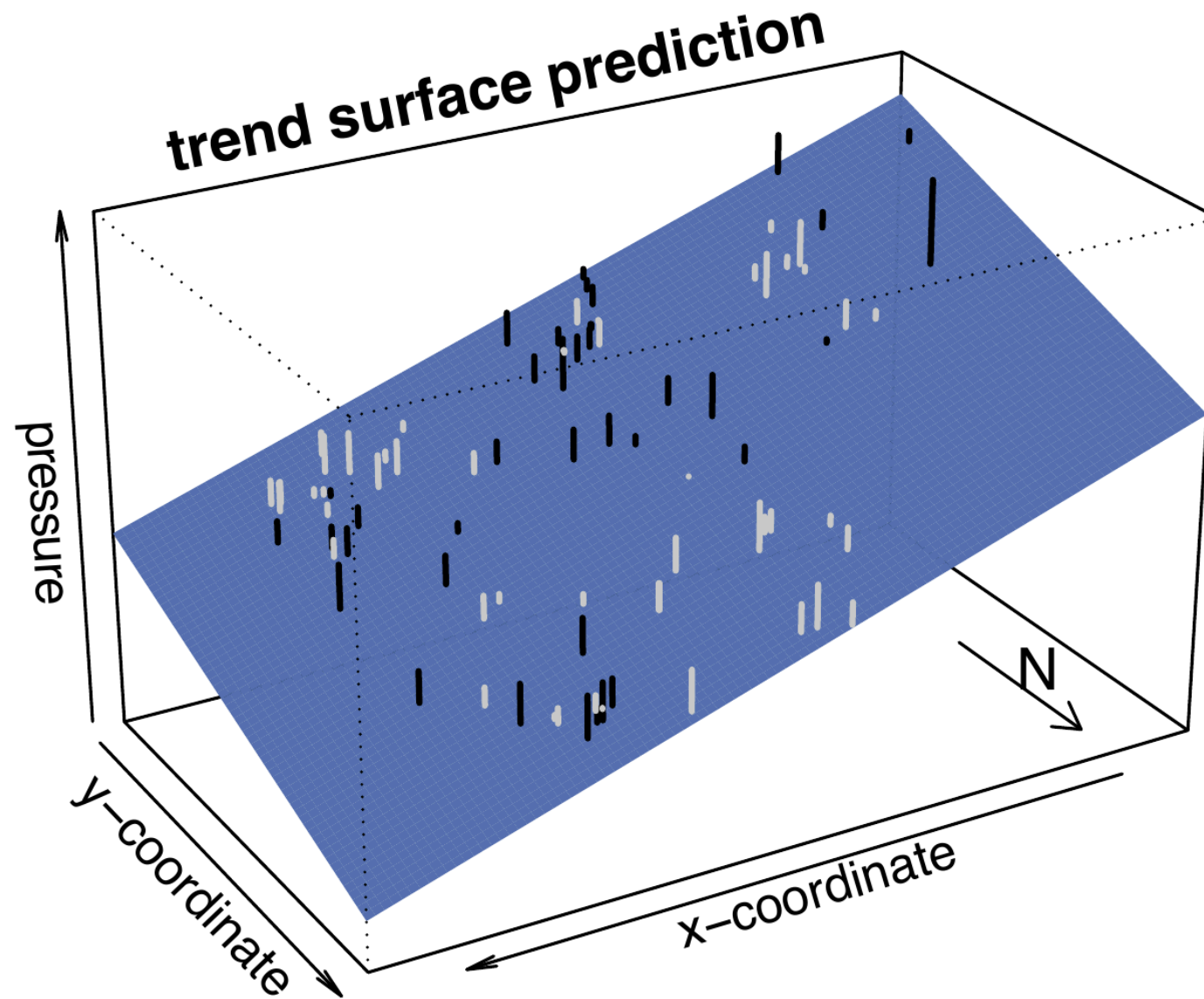


# Wolfcamp aquifer data

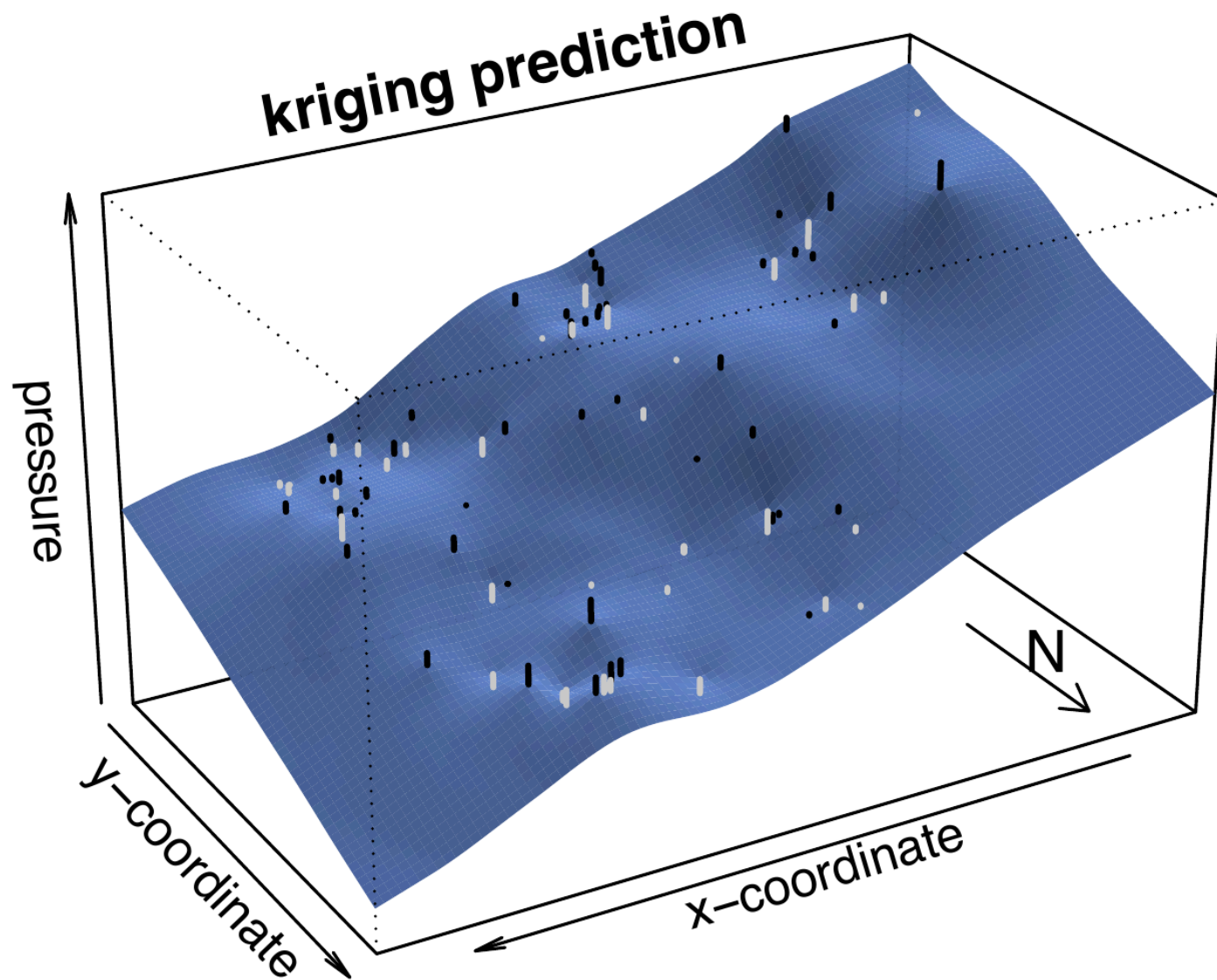
Note: 3D plot is rotated, North-East in the lower left corner



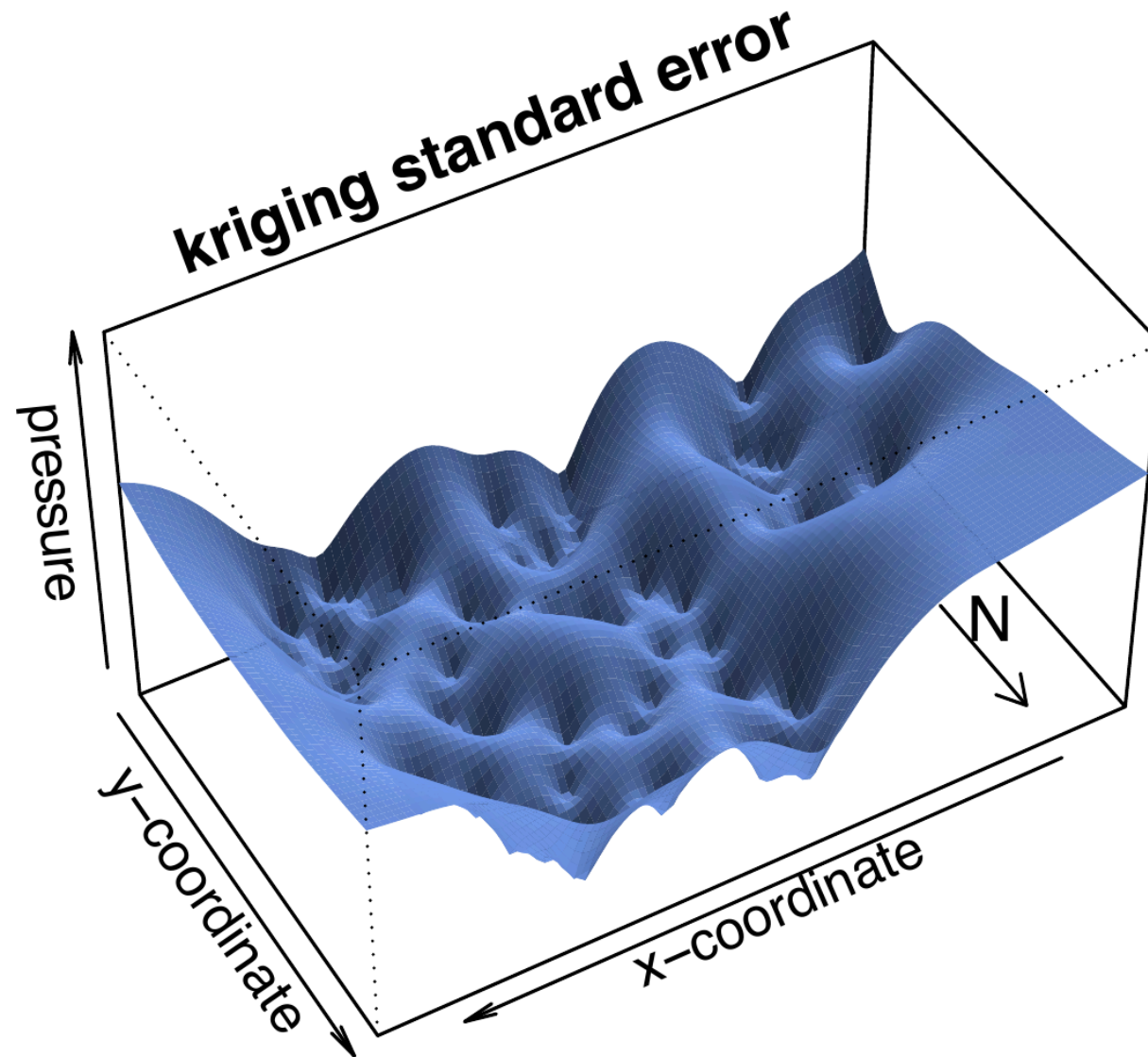
# Trend prediction



# Kriging prediction



# Kriging standard error



# 10 Summary

## Summary of the first geostatistical analysis

- modelling trend by linear regression model based on insights from exploratory analysis
- modelling residual auto-correlation by variogram
- simultaneous estimation of regression coefficients of trend model and parameters of variogram by (restricted) maximum likelihood
- hypothesis tests for spatial data should take auto-correlation into account



**Utrecht  
University**

Sharing science,  
*shaping tomorrow*