Spatial Statistics

Session 1

Madlene Nussbaum
m.nussbaum@uu.nl
Department of Physical Geography, Utrecht University

Andreas Papritz

07 Oct 2024

1 Overview:
Spatial data analysis

Definition

Spatial Statistics

Statistical analysis of data where the spatial (or spatio-temporal) position at which the attribute was recorded is known and relevant to the analysis.

Type of spatial data and their statistical sub-domain:

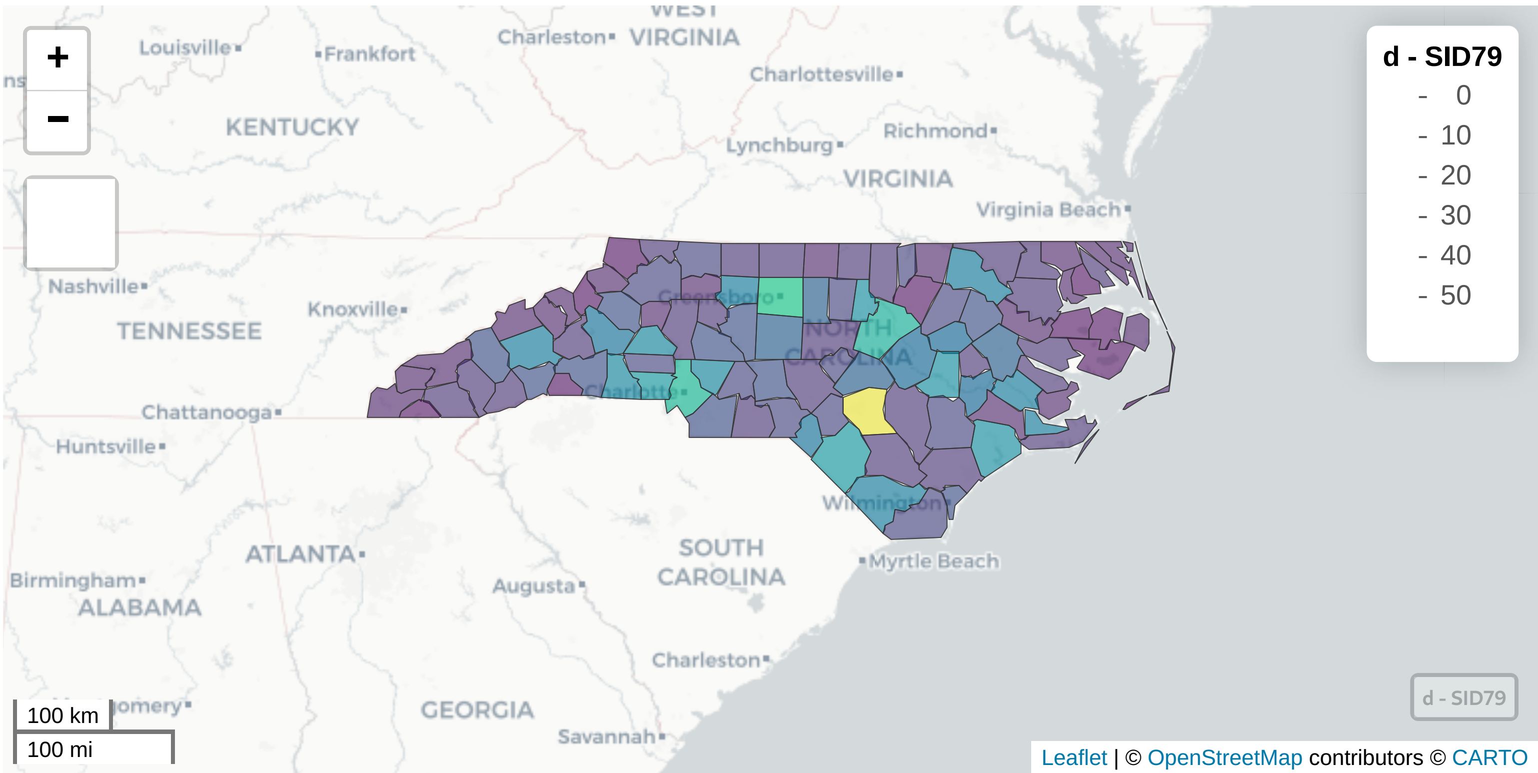
- **Areal data analysis**: analysis of spatial data that typically refer to larger areas (volumes) and that are arranged on regular or irregular lattices (with finite number of "cells")
- **Point pattern analysis**: analysis of spatial positions of "points" (or other objects) in a study domain
- **Geostatistics**: analysis of spatial data that refers to a very small area (volume) and that can in principle be recorded at any point in a study domain (⇒ infinite number of locations in study domain where measurements can be taken)

⇒ geostatistics is just one (important) branch of spatial statistics

Areal data: Polygon

In areal or lattice data, the domain D is a fixed countable collection of (regular or irregular) areal units at which variables are observed.

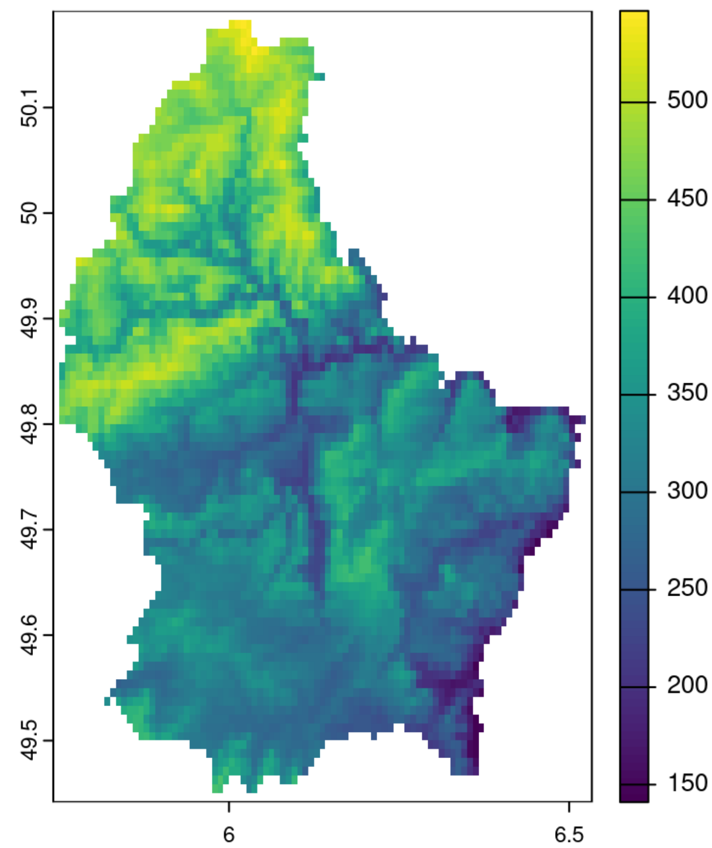
```r
library(sf)
library(mapview)
d <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
mapview(d, zcol = "SID79")
```



Number of sudden infant deaths per counties of North Carolina (USA) in 1979.

Areal data: Raster

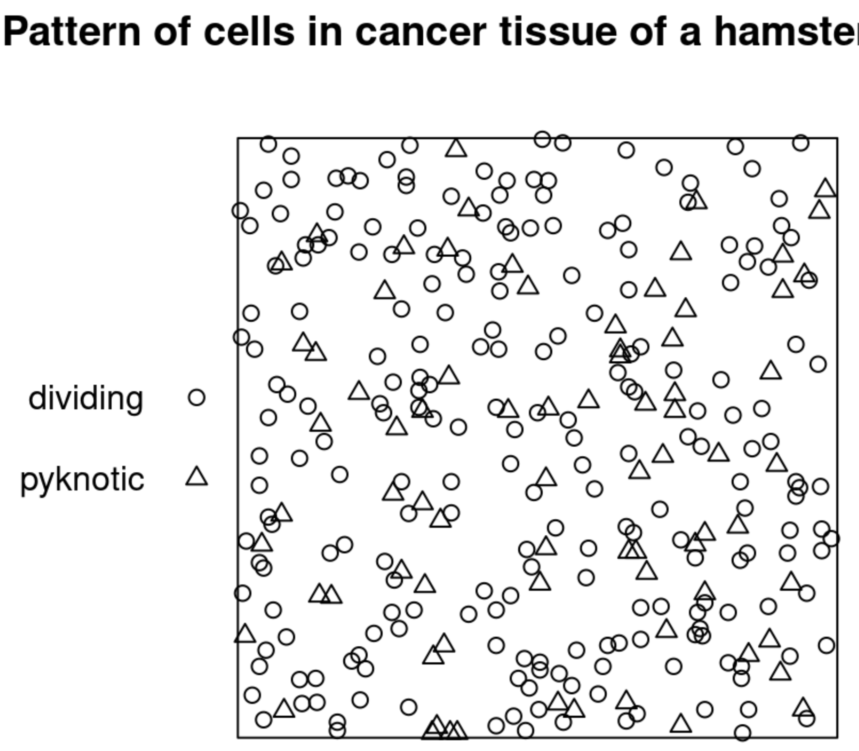
```r
library(terra)
d <- rast(system.file("ex/elev.tif", package = "terra"))
plot(d)
```



Elevation (m) at raster grid cells covering Luxembourg.

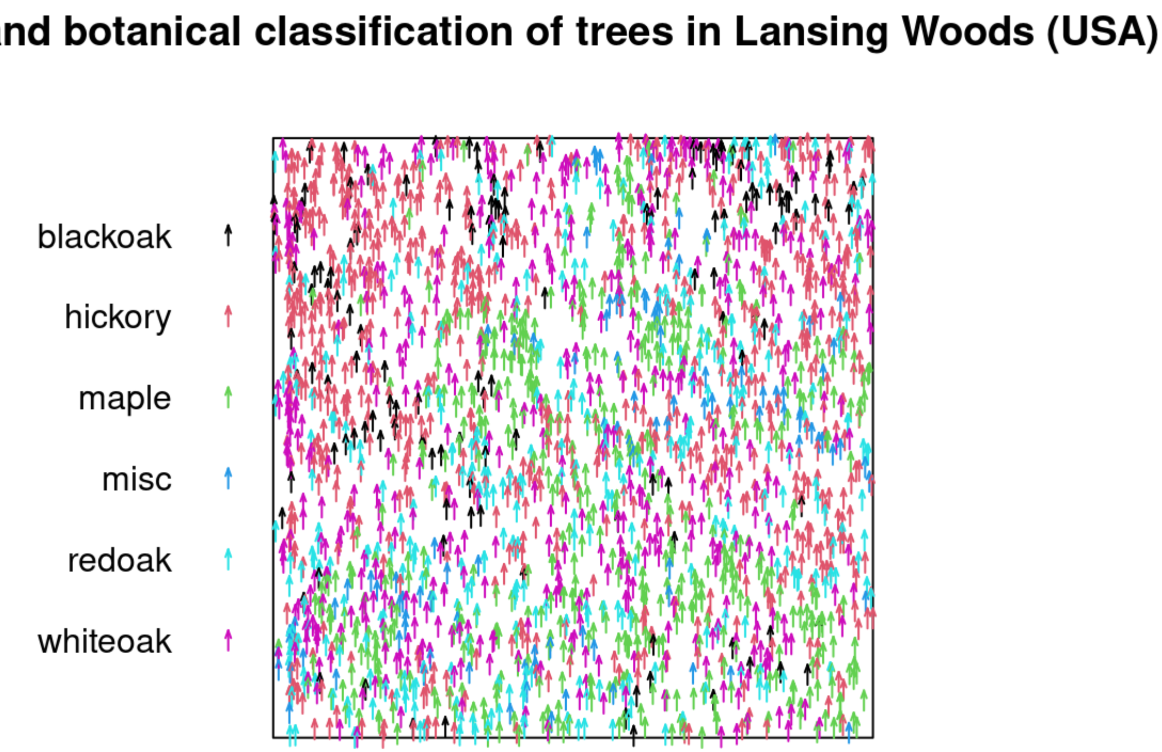
Point pattern

```r
library(spatstat)
plot(hamster, main = "Pattern of cells in cancer tissue of a hamster")
```



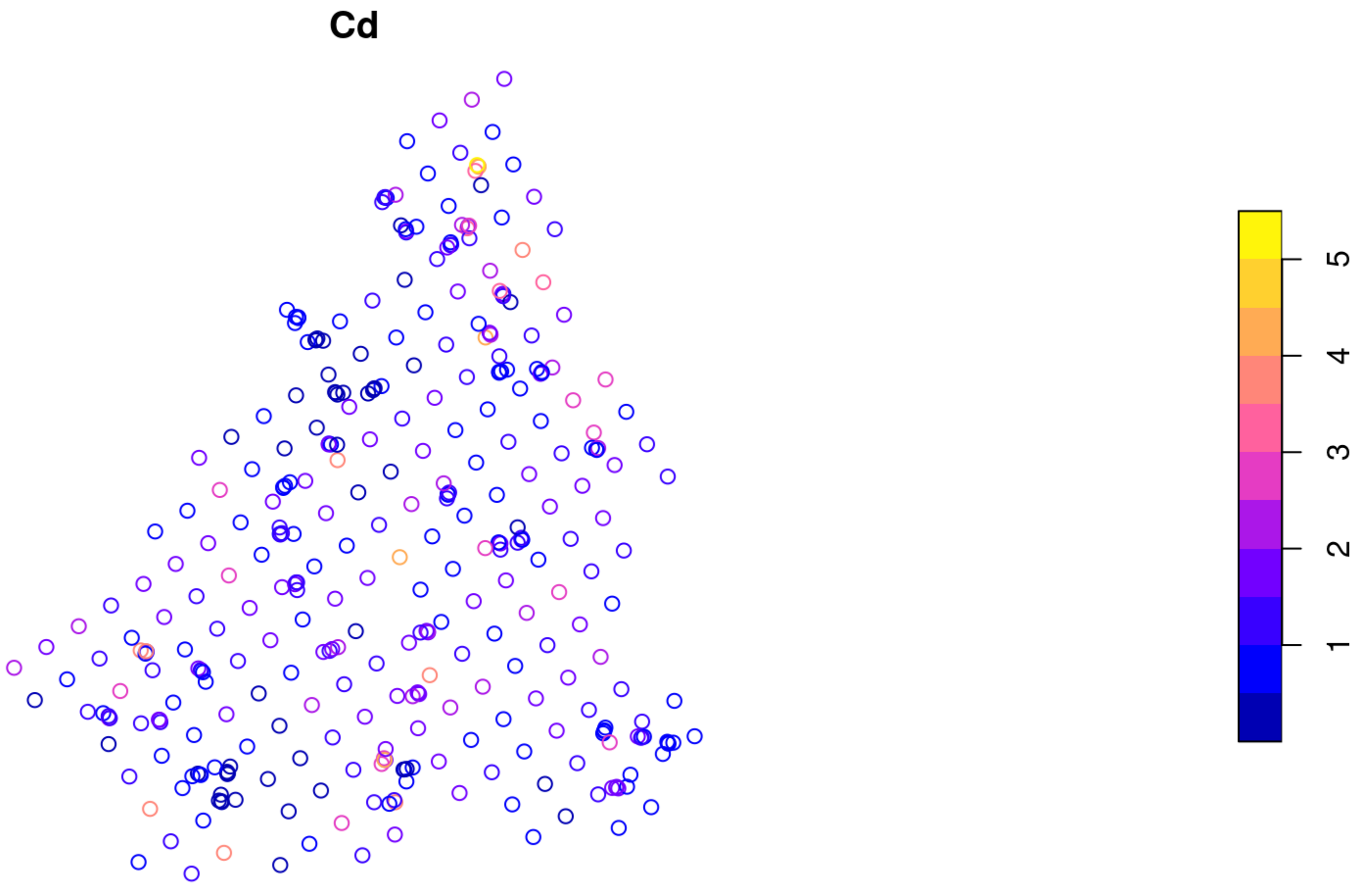
Point pattern

```
library(spatstat)
plot(lansing, shape="arrows", direction=90, cols=1:6, main = "Locations and botanical classificati
```



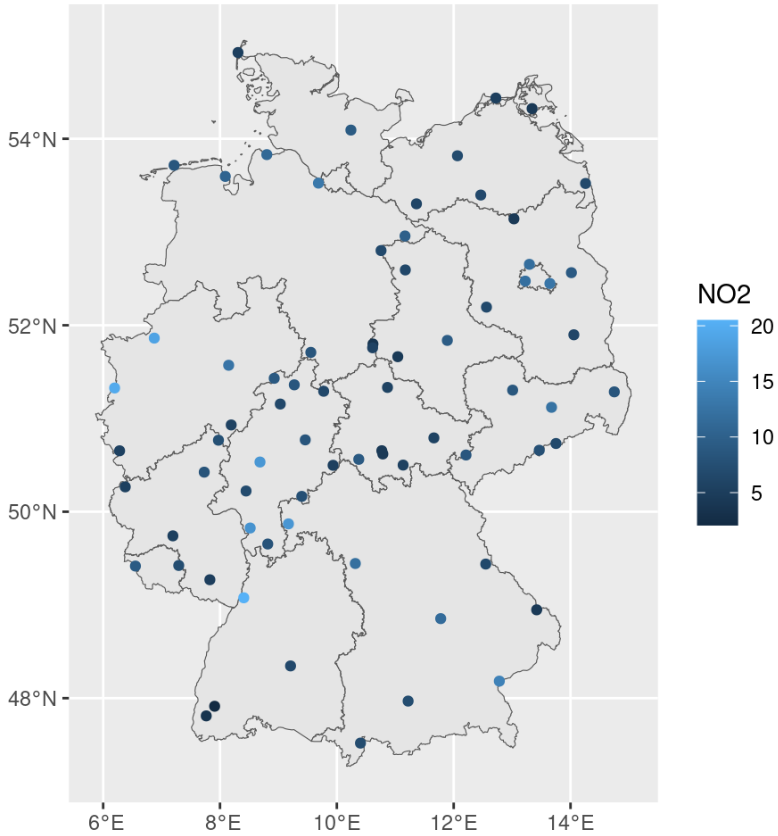
Geostatistical data: Soil pollution

```r
library(compositions)
data(juraset)
d <- st_as_sf(juraset, coords = c("X", "Y"))
plot(d[3])
```



Geostatistical data: Air pollution

```
library(gstat)
library(ggplot2)
no2 <- read.csv(system.file("external/no2.csv",package = "gstat"))
v.no2 <- st_as_sf(no2, crs = "OGC:CRS84", coords = c("station_longitude_deg", "station_latitude_de
v.de <- read_sf("https://github.com/edzer/sdsr/raw/main/data/de_nuts1.gpkg")
ggplot() + geom_sf(data = v.de) + geom_sf(data = v.no2, mapping = aes(col = NO2))
```



2 Geostatistical analysis

2.1 Objectives of geostatistical analyis

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

2.2 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 (⇒ trend surface) or as function of external spatial covariates (⇒ external trend)
  - **local fluctuations** usually spatially structured (values at pairs of nearby locations “more similar” than for pairs farther apart) ⇒ spatial auto-correlation
- Spatial data is sometimes distorted by independent measurement errors

2.3 Terminology and model notation

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

where

$S(\mathbf{x}_i)$: Y_i > i^{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 $\mu(\mathbf{x}_i)$

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

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

3 Example of a geostatistical analysis with R

3.1 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*

```
library(sp)
library(georob)
data(wolfcamp, package="georob")
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 ...
```

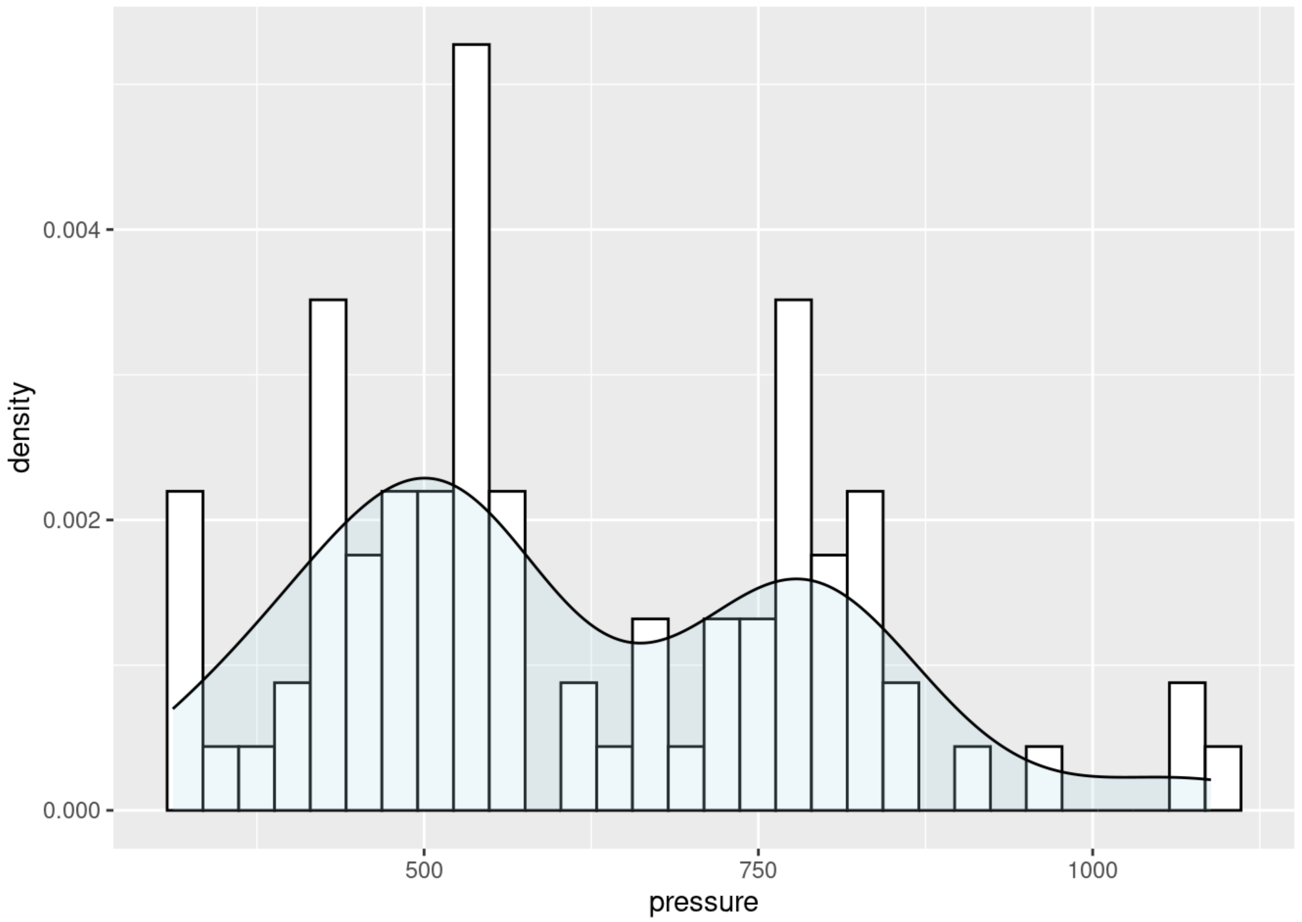
3.2 Exploratory analysis

```
# create sp object
d.w <- st_as_sf(wolfcamp, coords = c("x", "y"))
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

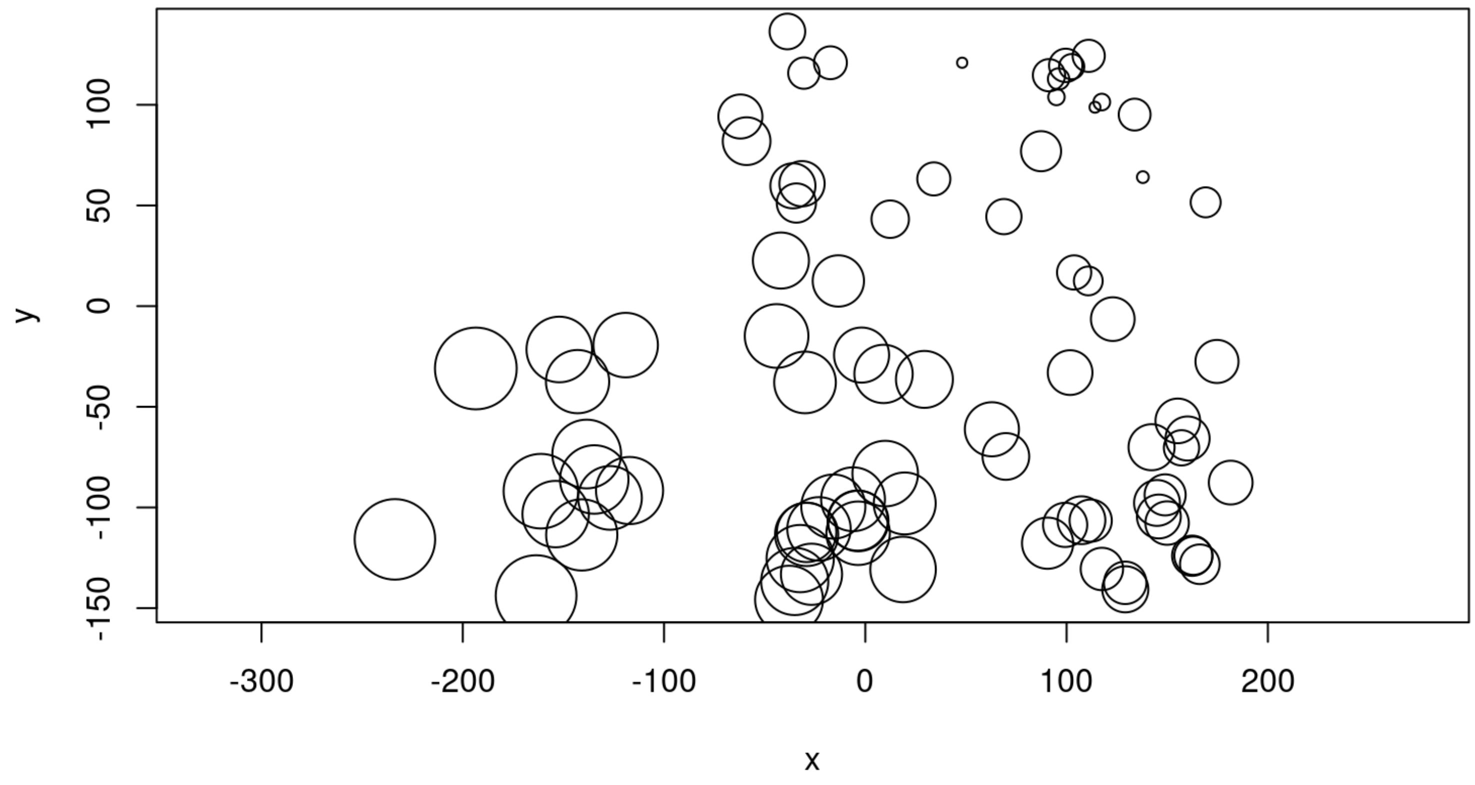
```
library(ggplot2)
ggplot(wolfcamp, aes(x = pressure)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="lightblue")
```



Spatial distribution of values

Bubble plot: symbol area linearly related to data

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



Spatial distribution of values

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

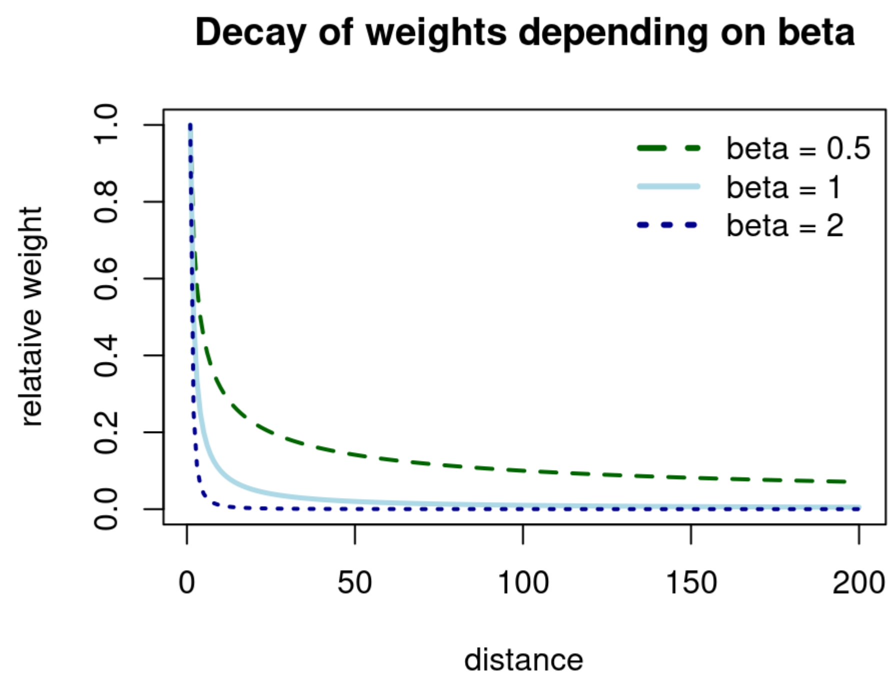
3.3 Ad-hoc approach: Inverse distance weighting

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 β:

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

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

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



Inverse distance weighting implementation

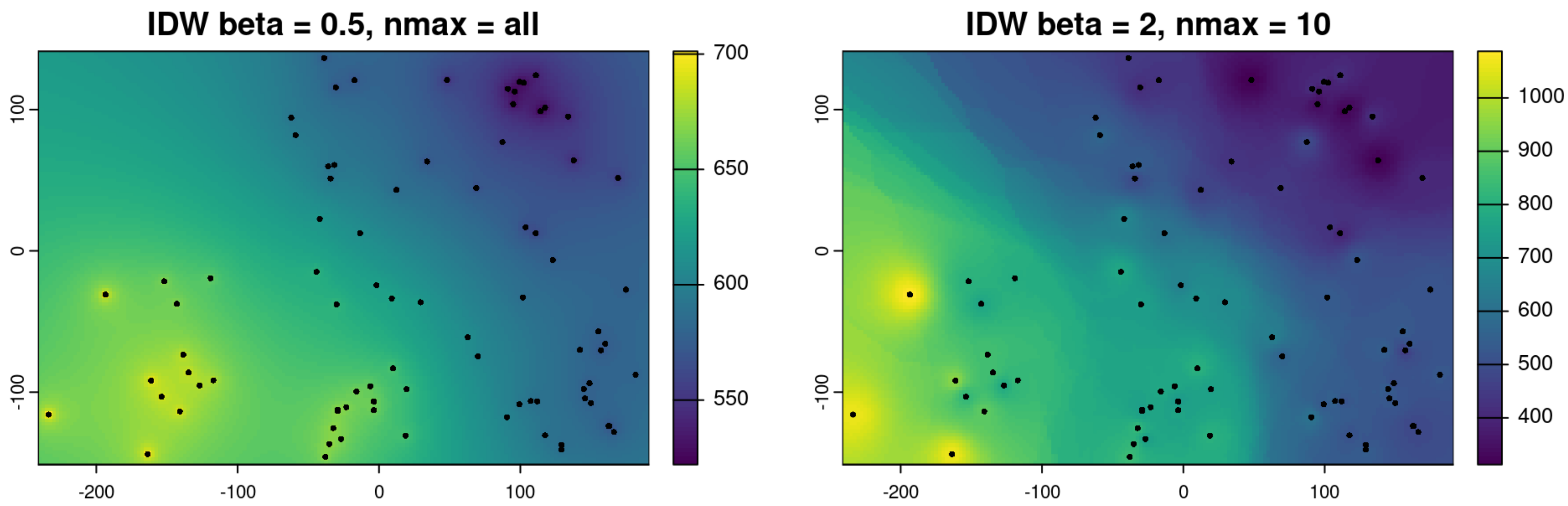
```r
library(gstat)
library(terra)
idw <- gstat(id = "pressure", formula = pressure ~ 1, data = d.w,
             nmax = nrow(d),        # use all the neighbors locations
             set = list(idp = 0.5)) # decay of weights by d^0.5

# create a regular grid
d.w.grid <- expand.grid(
  x = seq(-240, 190, by= 2.5),
  y = seq(-150, 140, by= 2.5)
)
gridded(d.w.grid) <- ~x+y

# predict for each node of grid
d.w.idw <- predict(idw, newdata= d.w.grid)
```

```
[inverse distance weighted interpolation]
```

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

```
class       : SpatRaster
dimensions  : 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
```

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



Once we found optimal β, is our job done?!

Discuss …

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

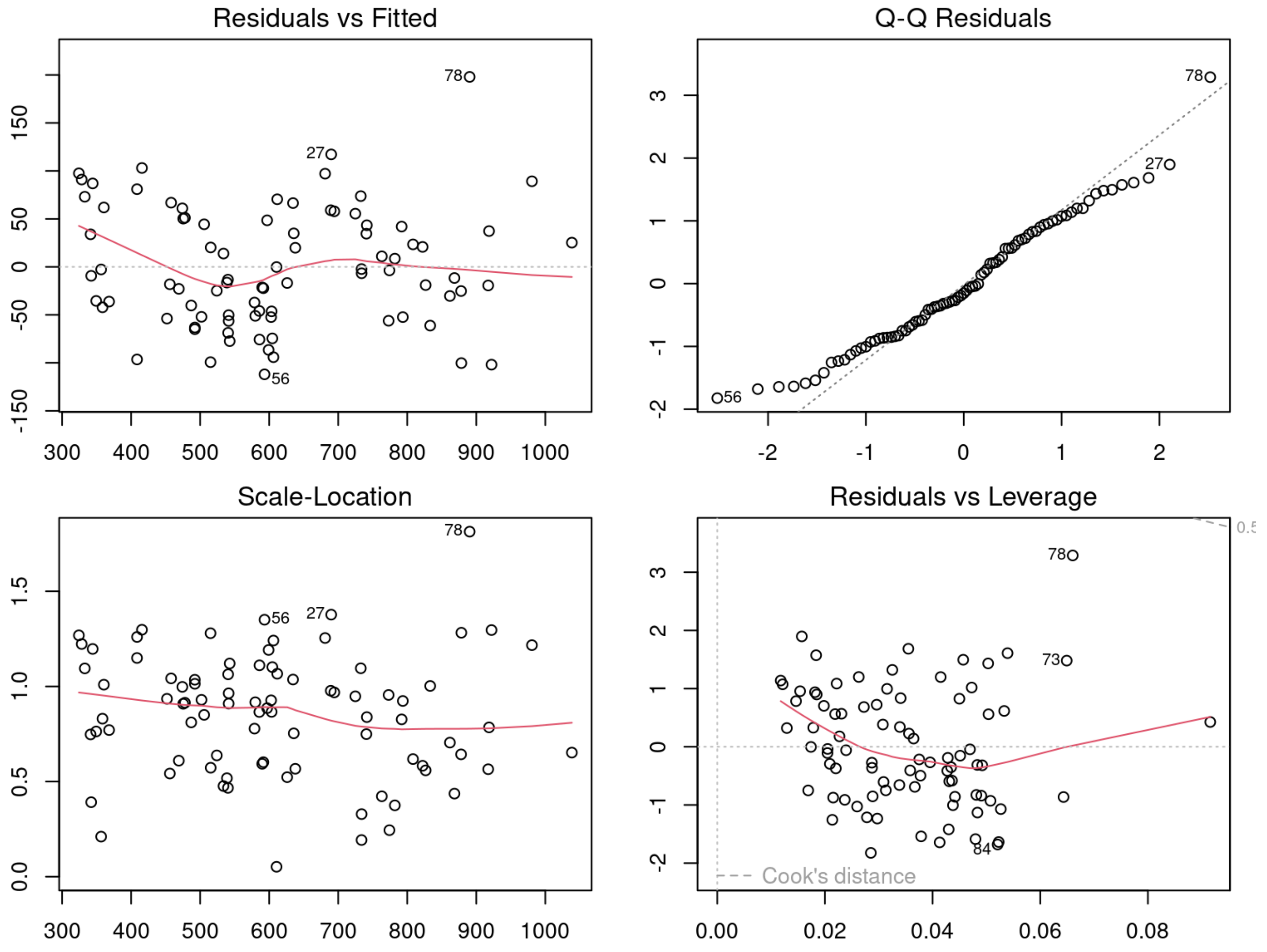
3.4 Trend modelling

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

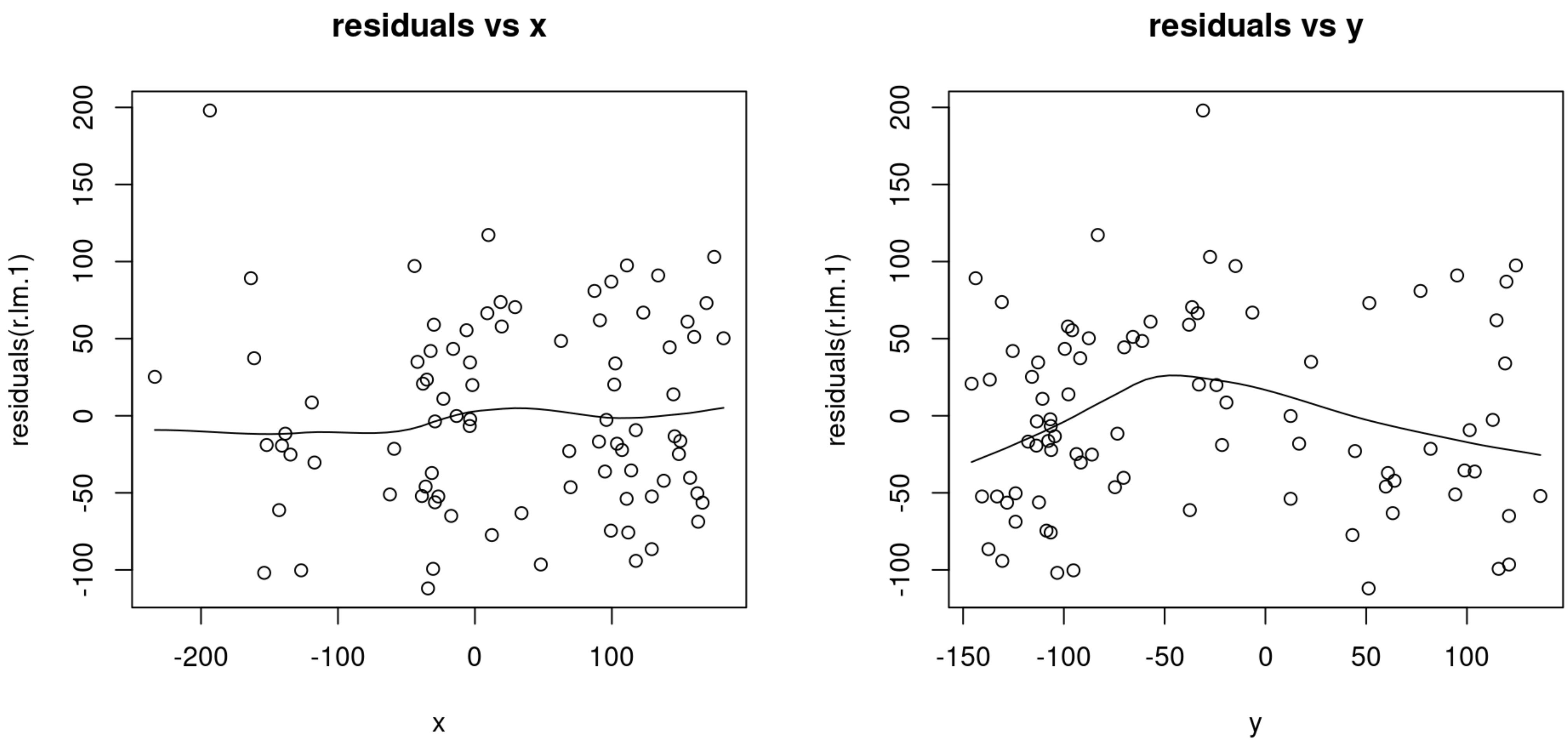
```
Call:
lm(formula = pressure ~ x + y, data = wolfcamp)

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

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



Residuals plotted against spatial coordinates



Include higher order polynomials

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

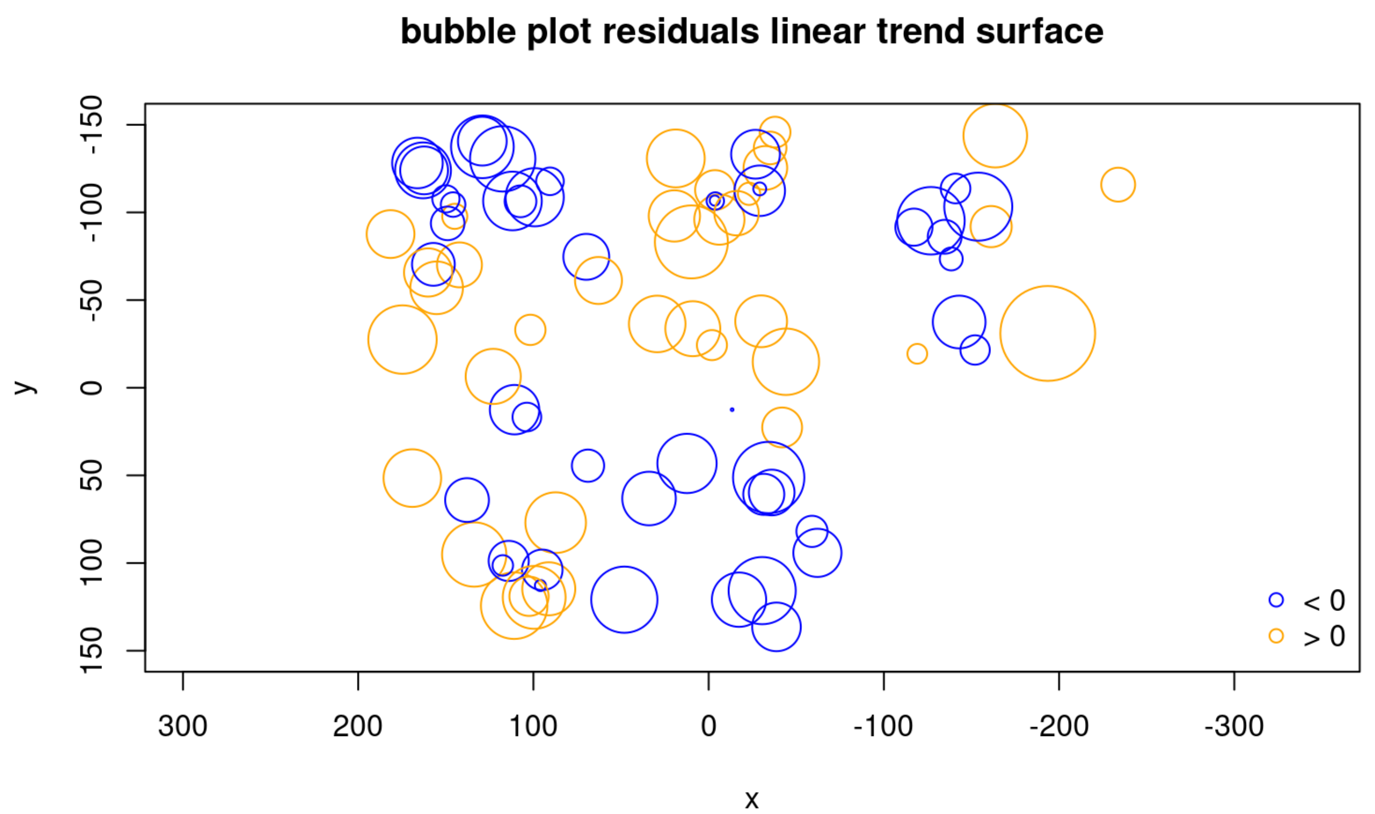
Compare models

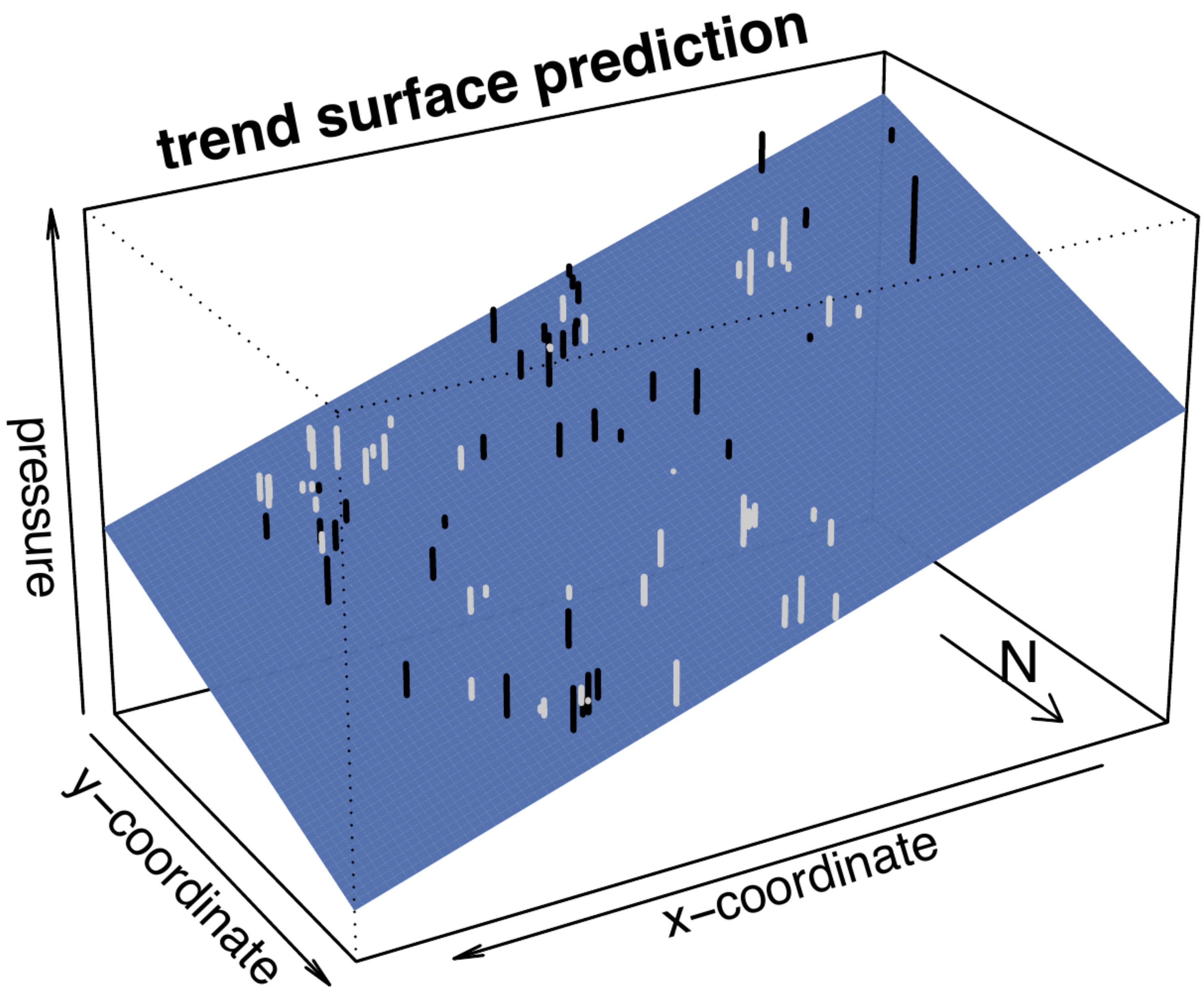
```
anova(r.lm.1, r.lm.2)
```

```
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

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




trend surface prediction
pressure
y–coordinate
x–coordinate
N

3.5 Estimating and modelling auto-correlation

Pro memoria: sample covariance and correlation

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

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

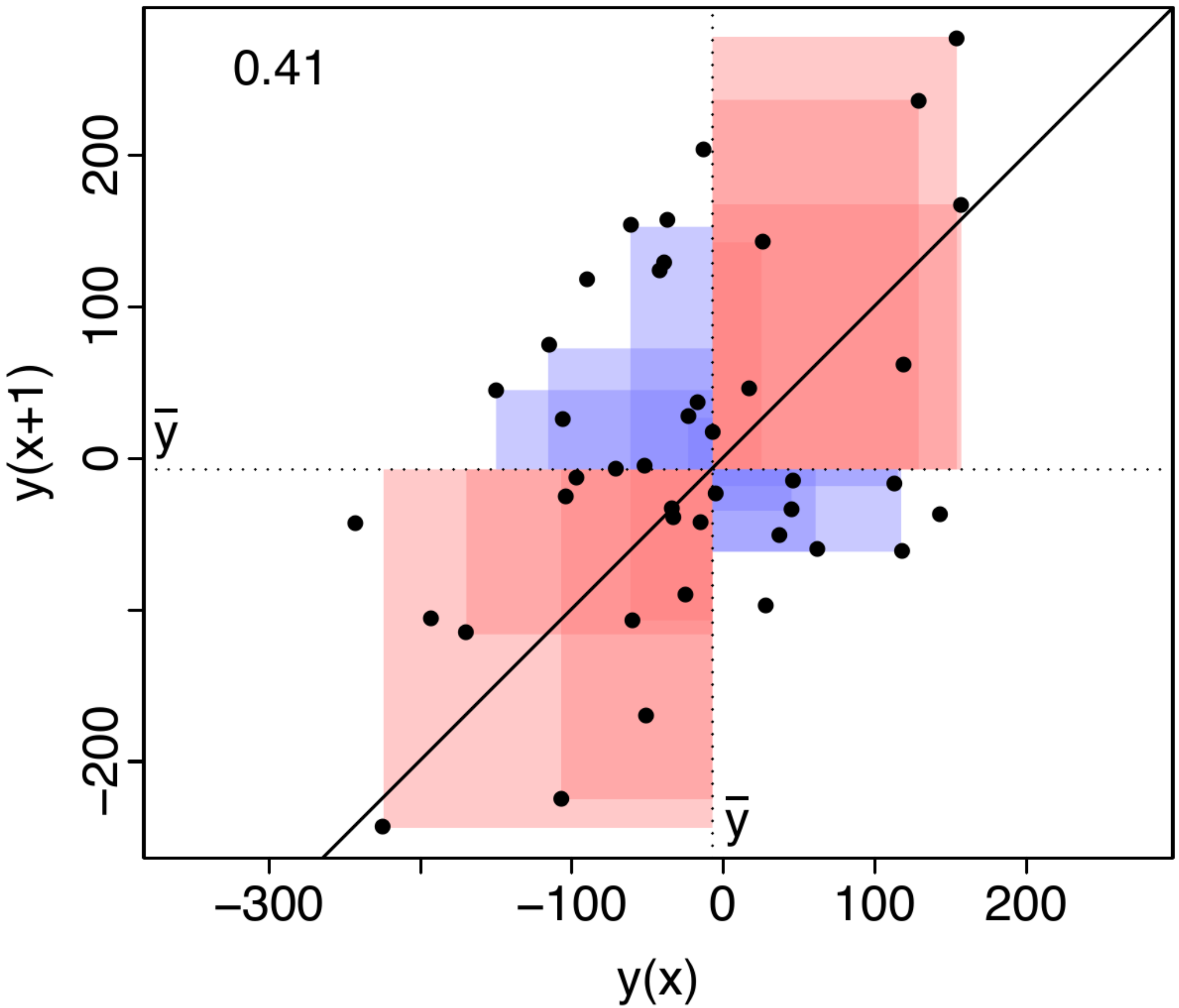
- (Pearson) correlation coefficient, where s_1 and s_2 are the sample standard deviations

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

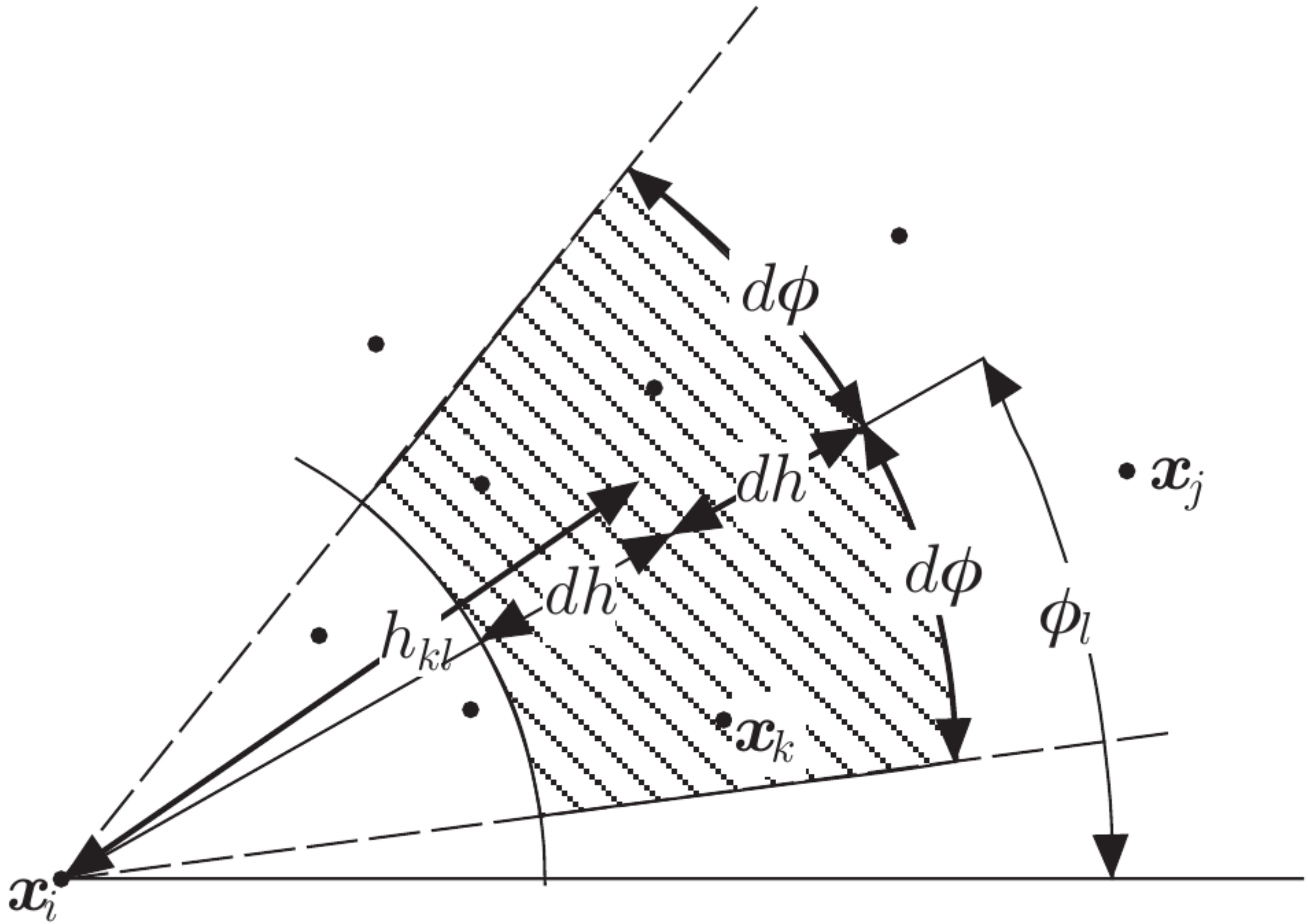
- “plug-in” estimator for auto-correlogram of time series

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

Covariance and correlation: visual display



Defining lags for irregular sampling grids



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

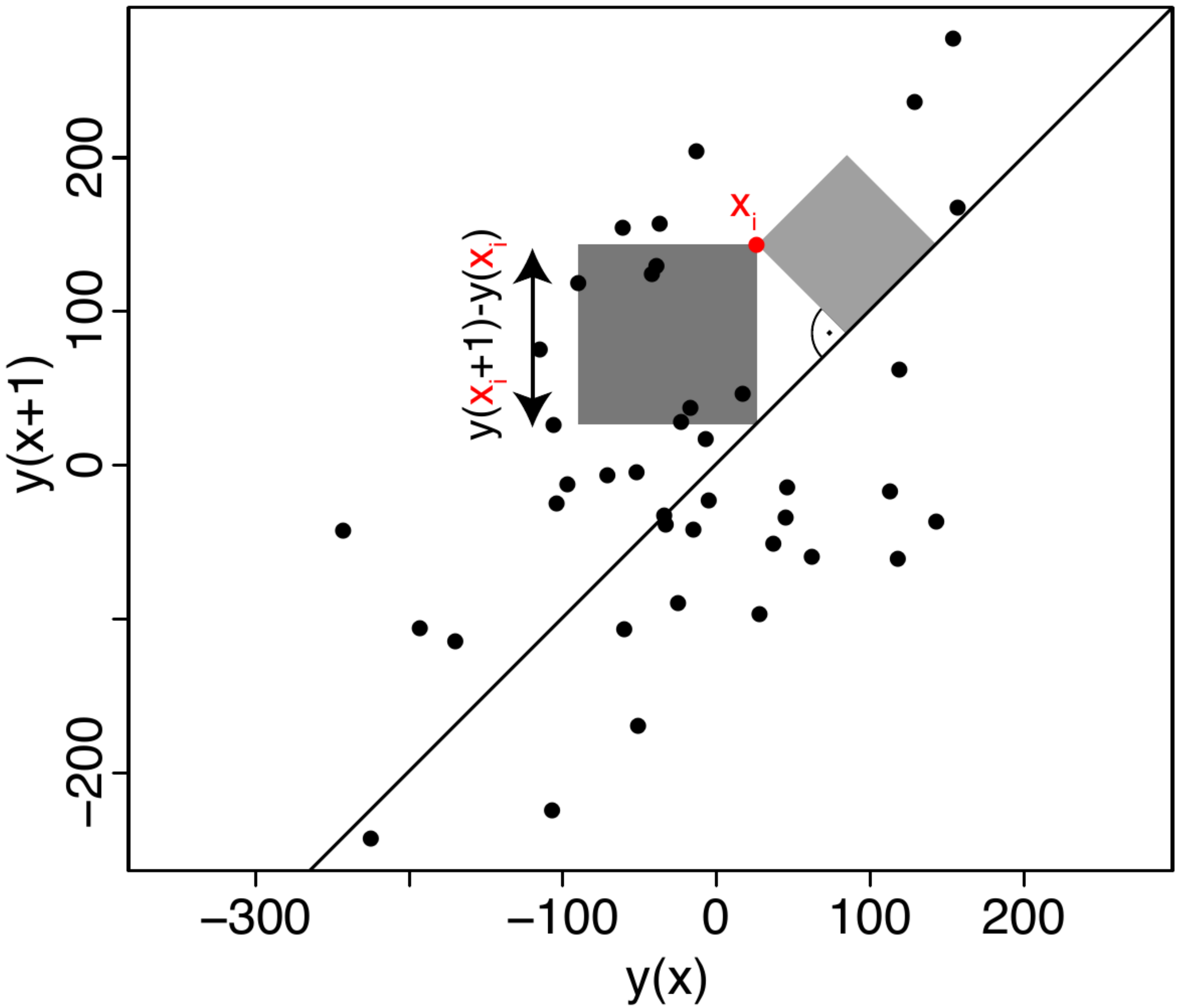
- $(k,l)^{\mathrm{th}}$ lag class, $\mathbf{h}_{kl}$, characterized by distance, $(h_k - dh, h_k + dh]$, and angular class, $(\phi_l - d\phi, \phi_l + d\phi]$
- N_{kl}: number of pairs of locations $(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathbf{x}_j - \mathbf{x}_i \approx \mathbf{h}_{kl}$
- **Covariogram**: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

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

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

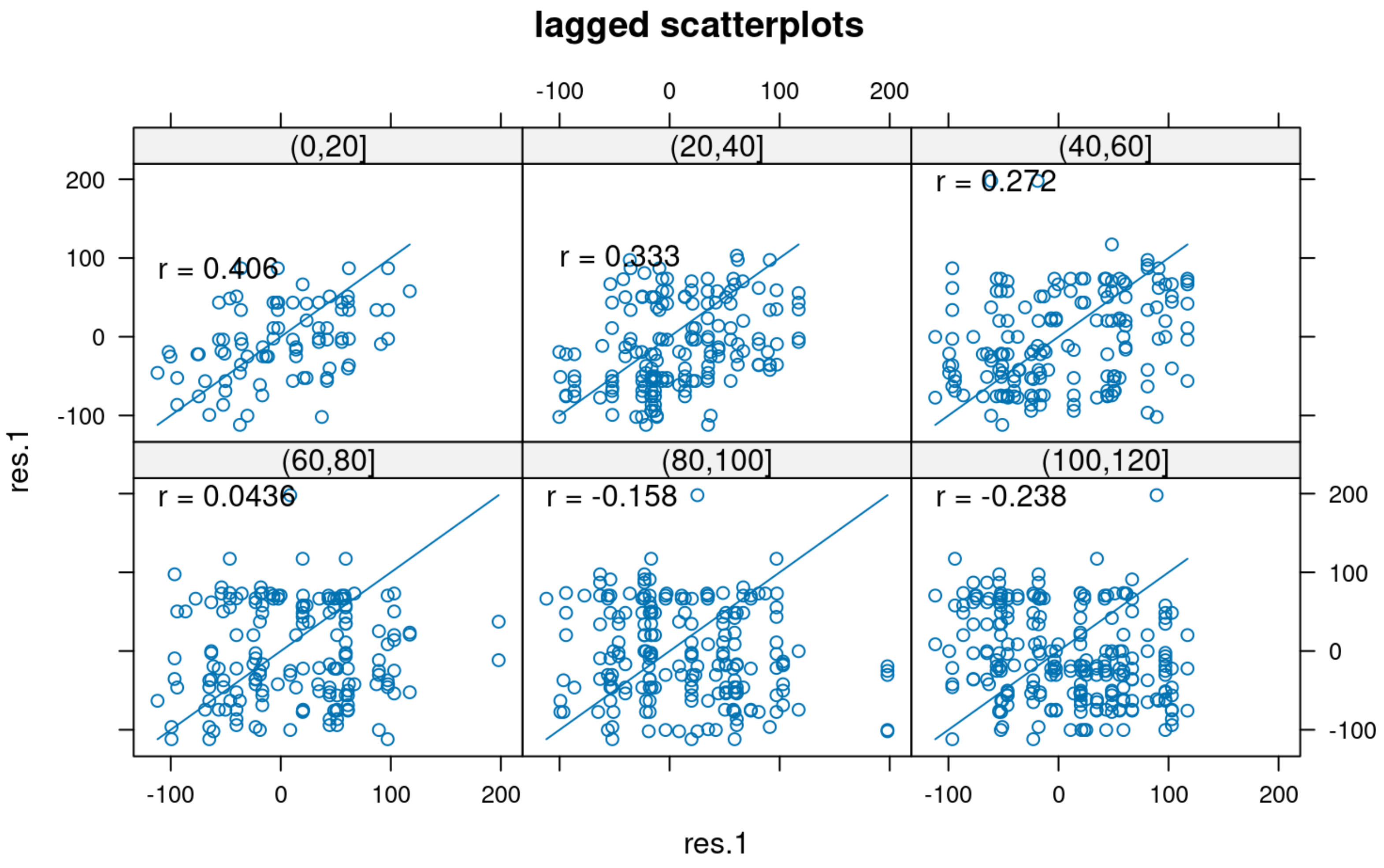
$$\widehat{V}(\mathbf{h}_{kl}) = \frac{1}{2\,N_{kl}} \sum_{(i,j)\in\mathbf{h}_{kl}} [y(\mathbf{x}_i) - y(\mathbf{x}_j)]^2$$

Spatial Auto-correlation: Semi-variance



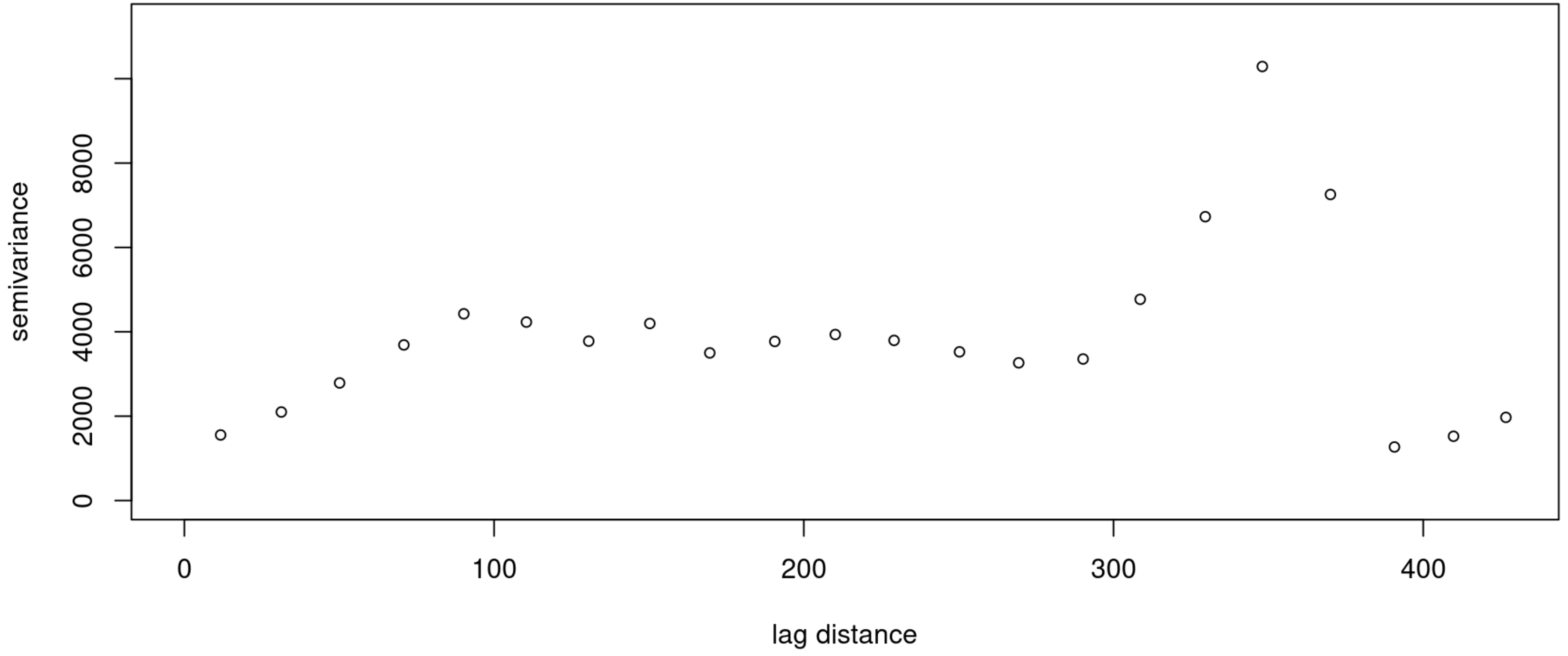
Lag-scatter plots of trend surface residuals

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

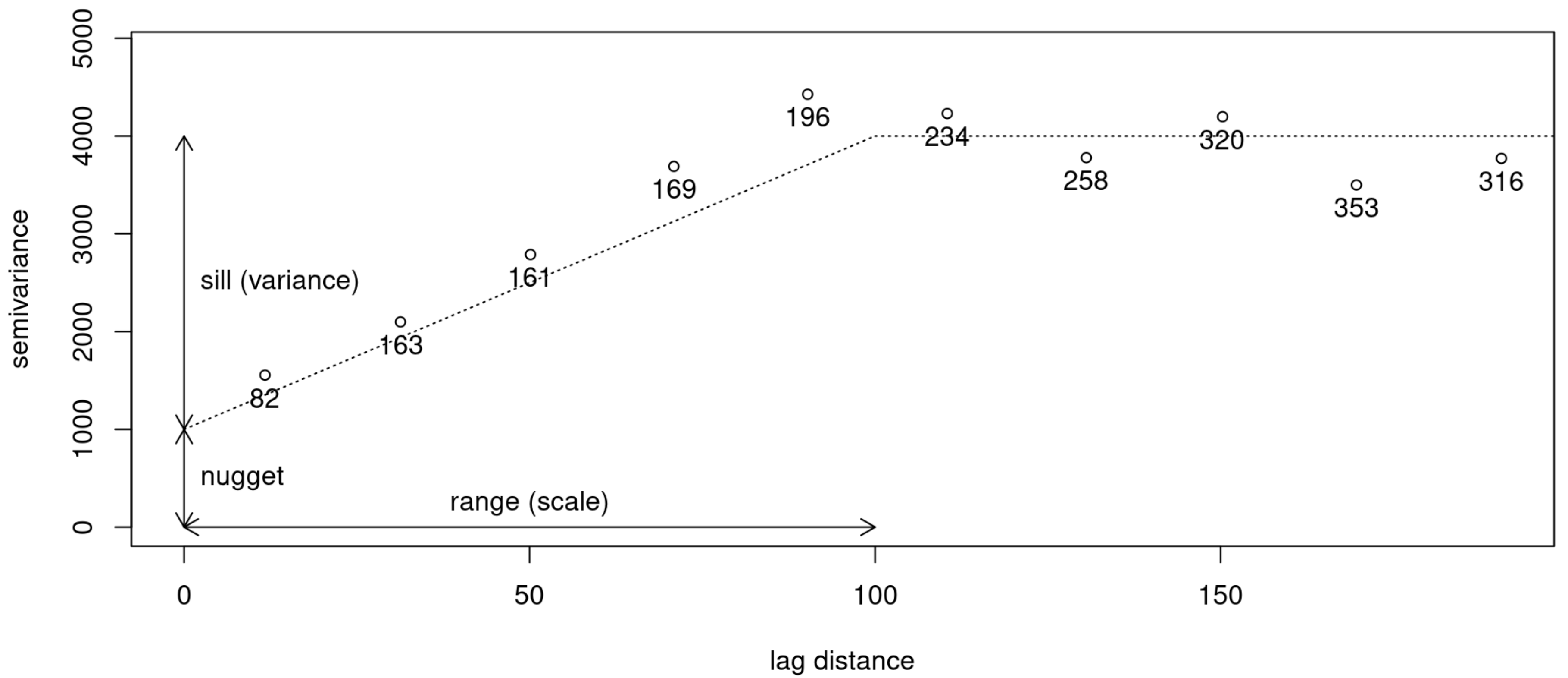


Variogram of trend surface residuals

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

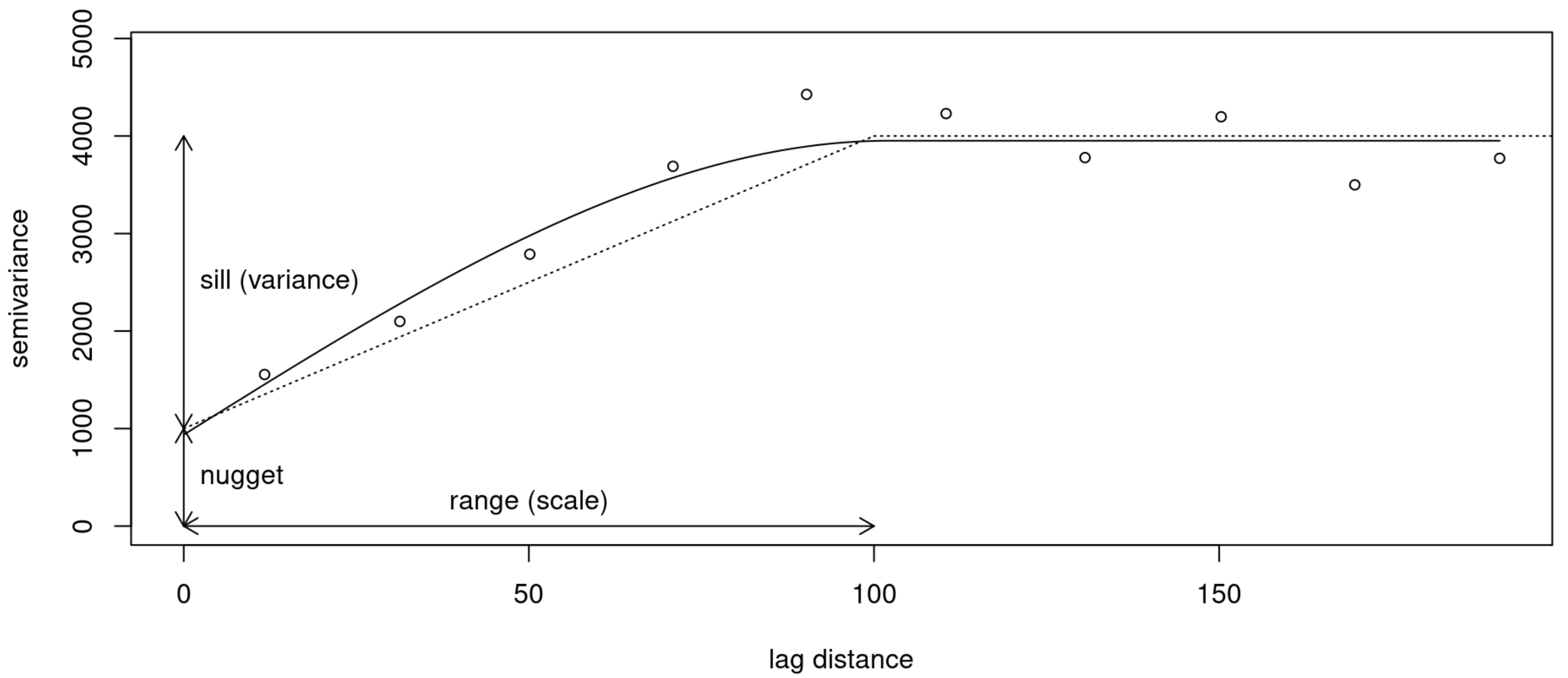


```
r.v <- sample.variogram(residuals(r.lm.1),
                        locations=st_coordinates(d.w),
                        lag.dist.def=20,
                        max.lag=200,
                        estimator="matheron")
plot(r.v)
text(gamma~lag.dist, r.v, labels=npairs, pos=1)
```



Fit variogram function

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



3.6 Fitting spatial model by maximum likelihood

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

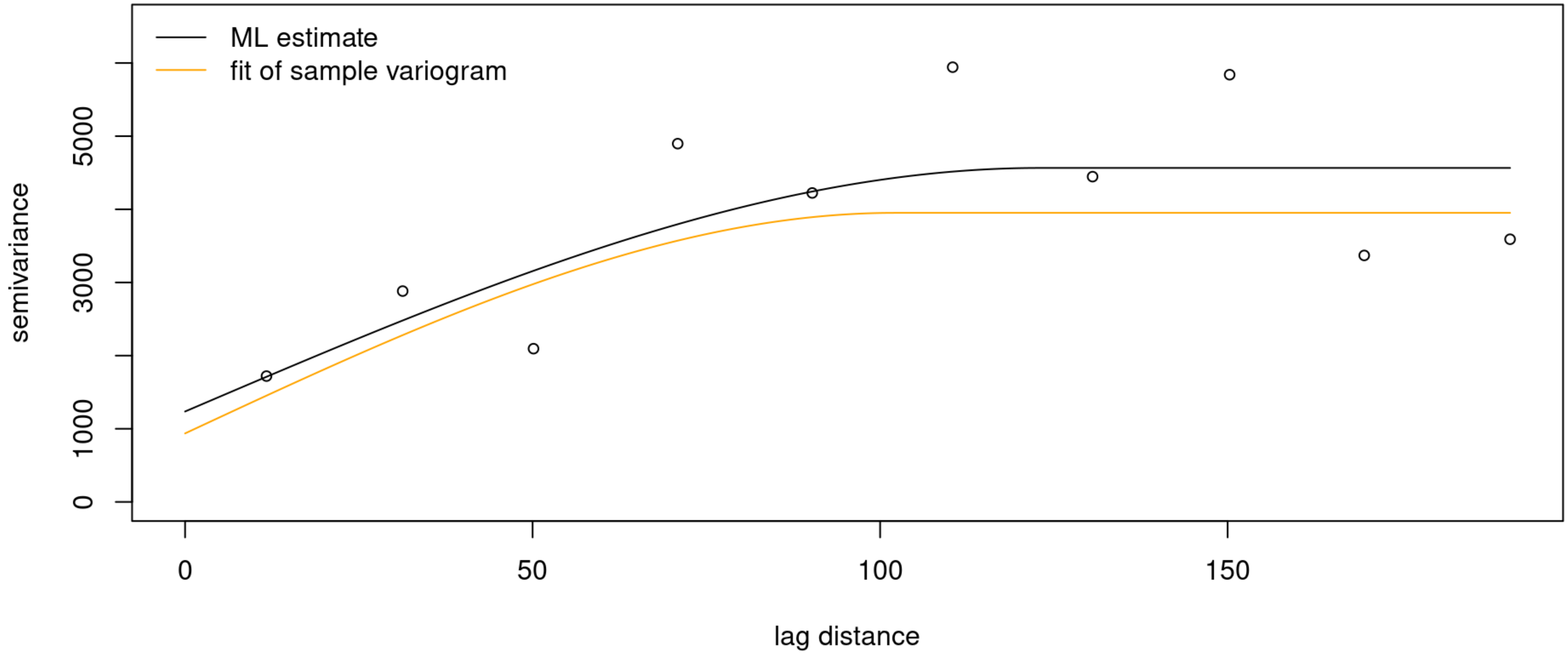
```
...
Maximized log-likelihood: -458.3671
...
Variogram:  RMspheric
                Estimate   Lower   Upper
variance         3328.90 1453.95 7621.7
snugget(fixed)      0.00      NA     NA
nugget           1236.27  616.01 2481.1
scale             122.95   95.13  158.9

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

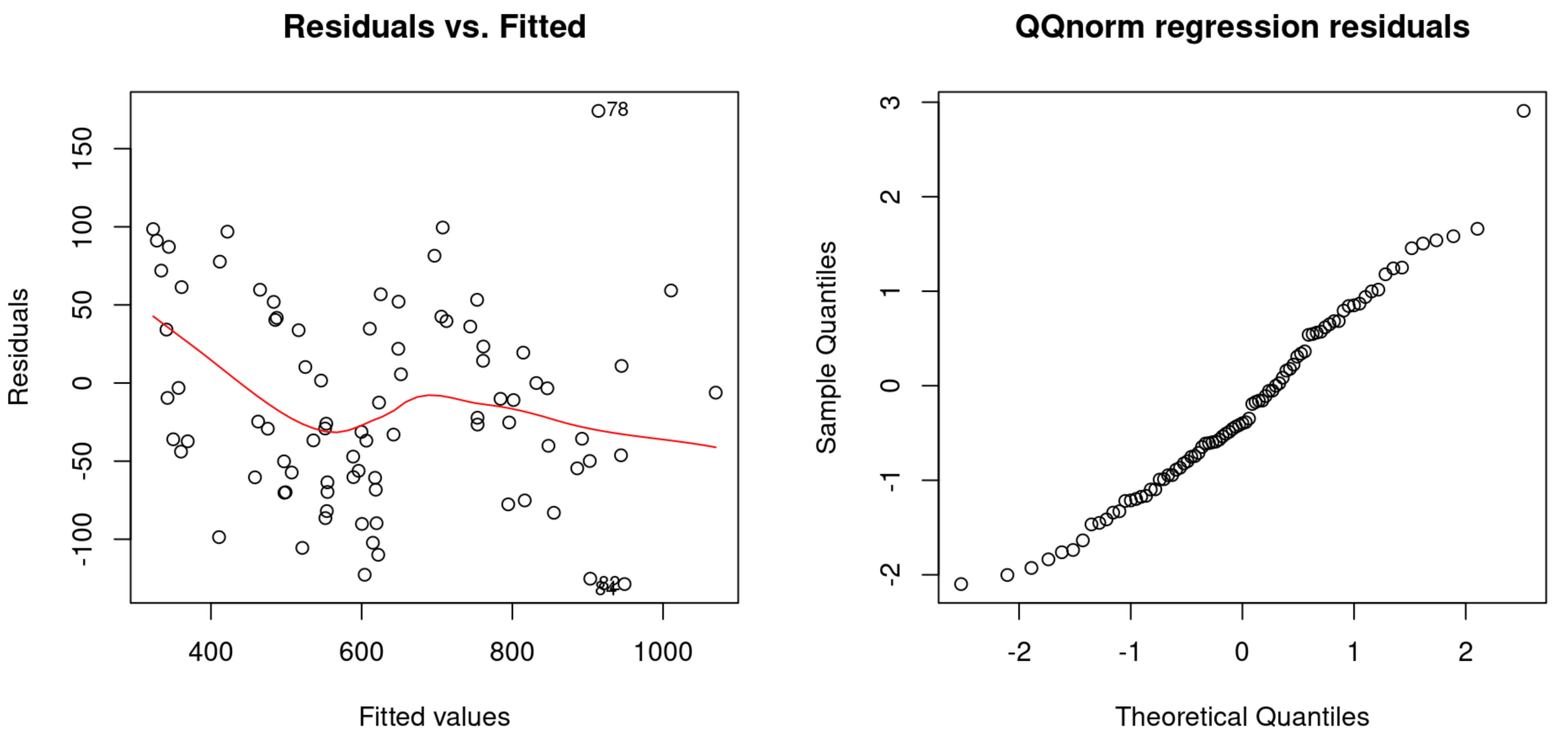
```r
summary(r.lm.1)
```

```

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
```

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



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



3.7 Inference, 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

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

```r
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
```

```r
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
```

3.8 Cross-validating trend surface models

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

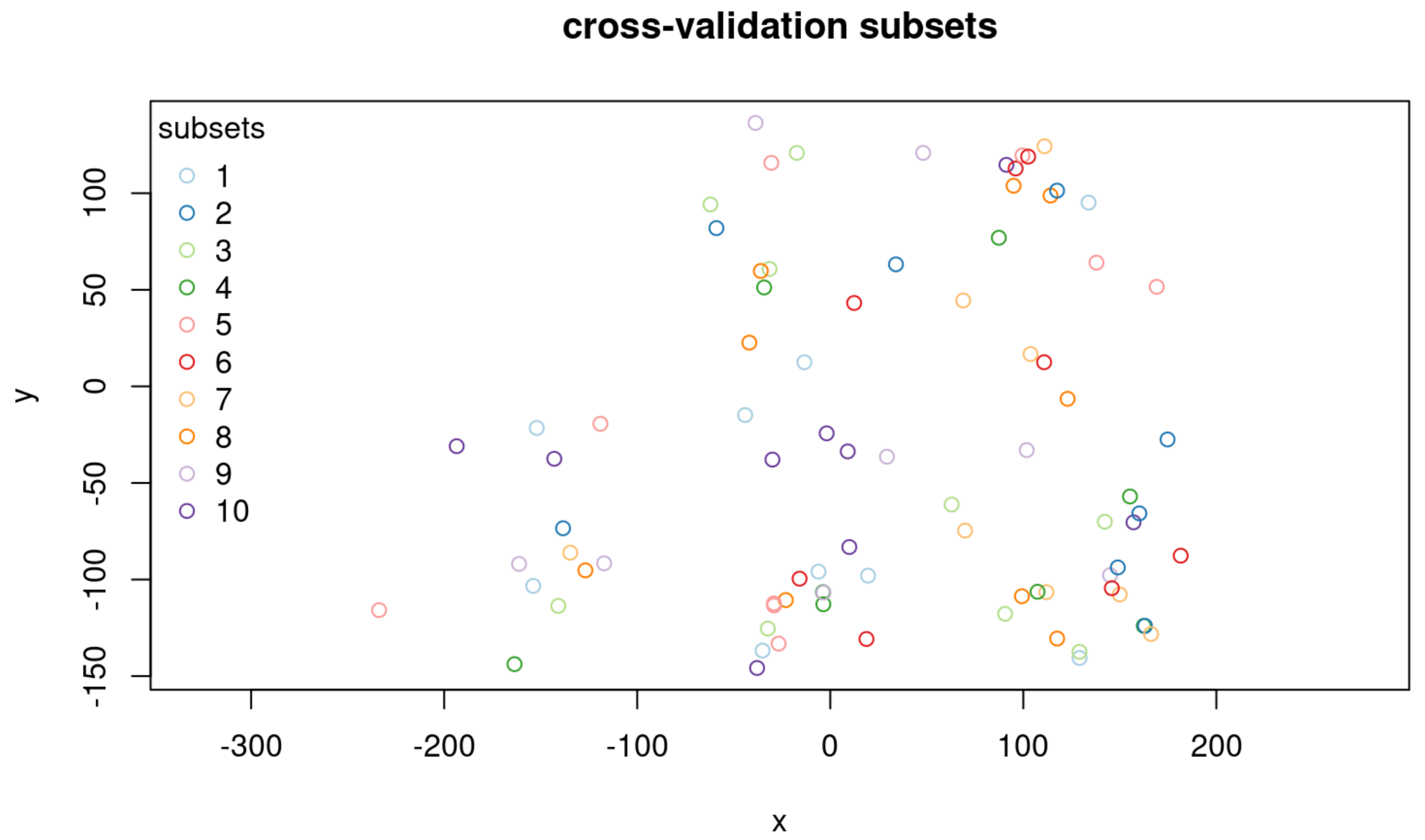
```r
summary(r.cv.1)
```

```
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 
```

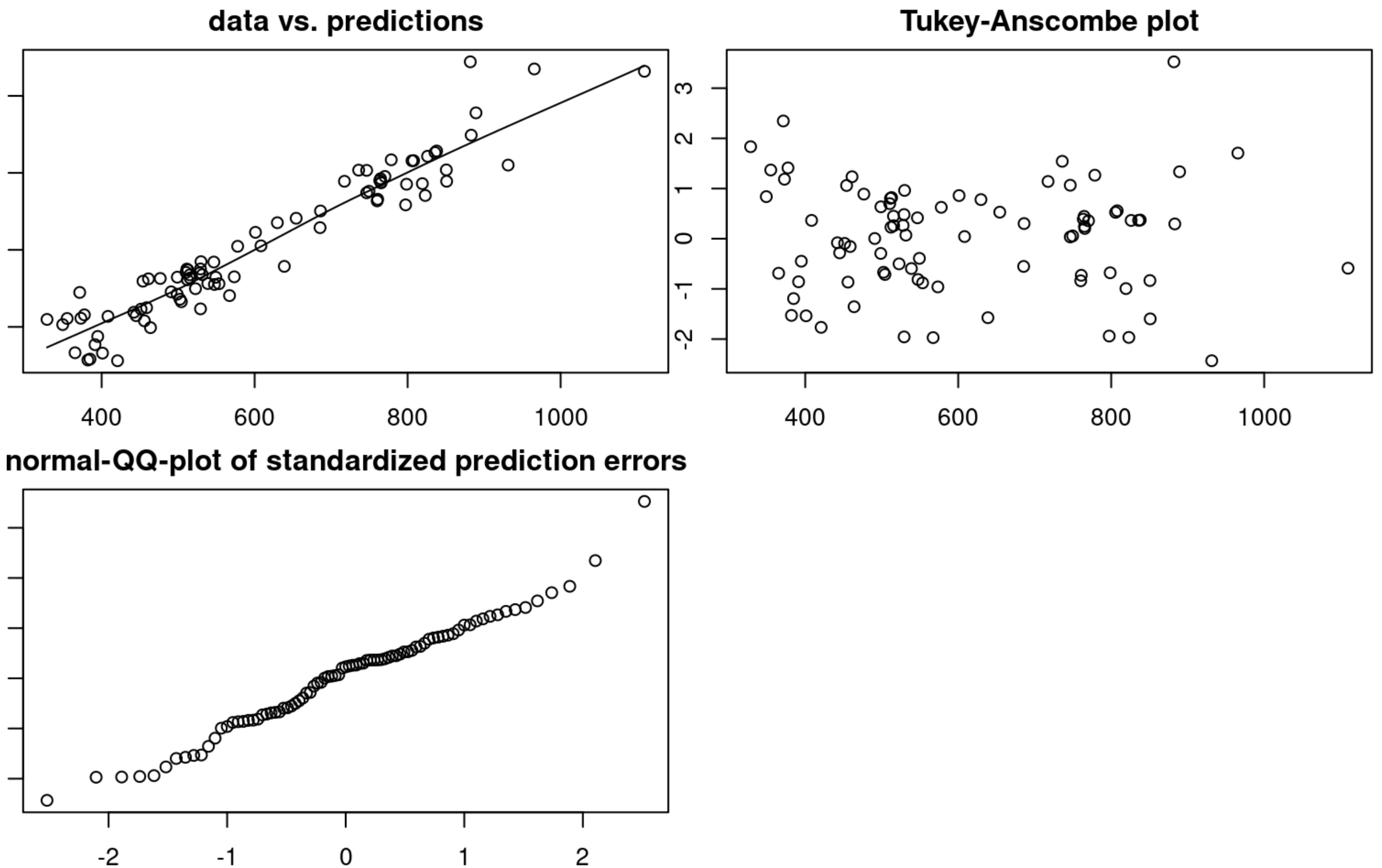
```r
summary(r.cv.2)
```

```
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 
```

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



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



3.9 Computing 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
- Item kriging provides in addition an estimate of the variance of the prediction error $S(\mathbf{x}_0) - \hat{S}(\mathbf{x}_0)$

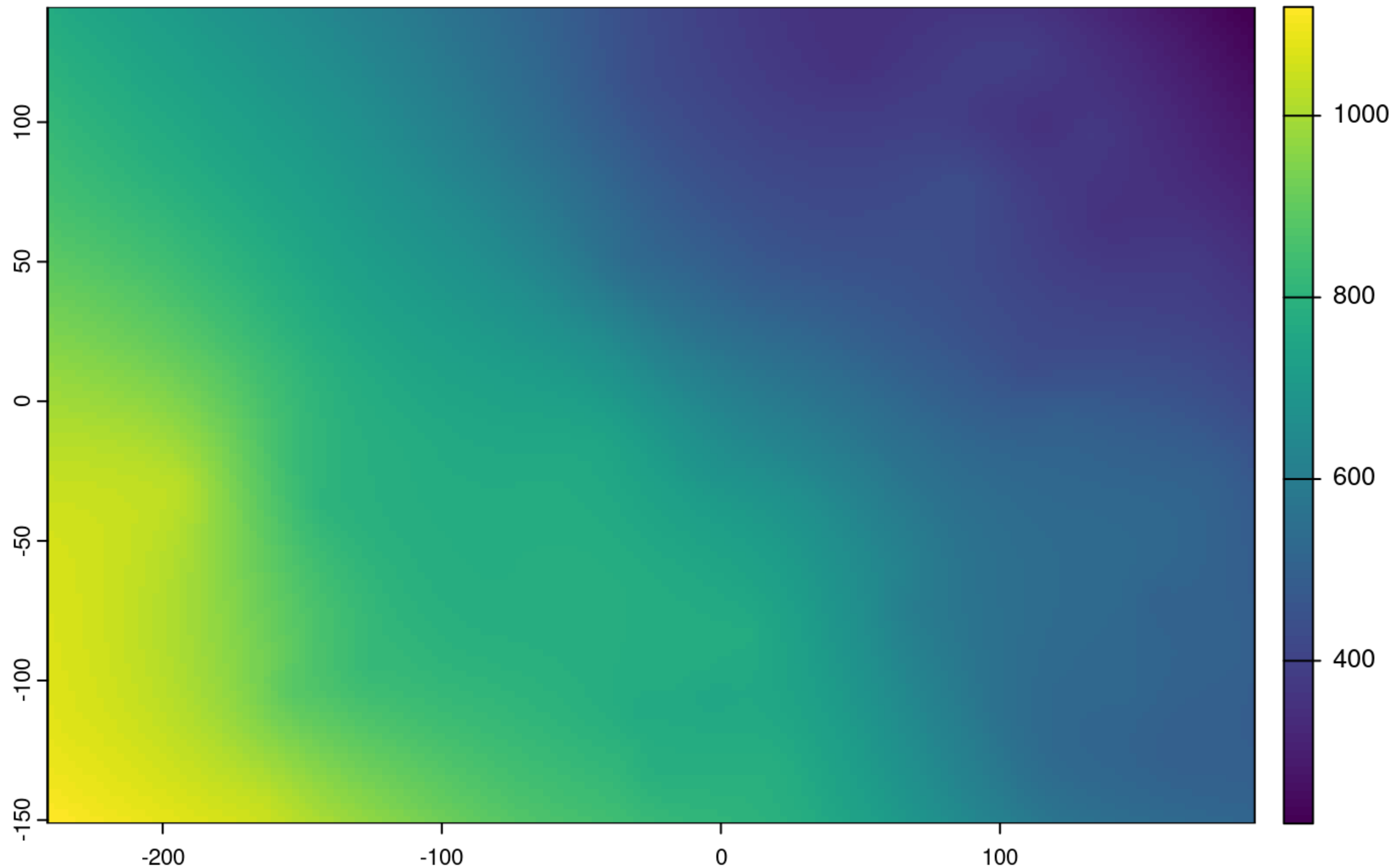
Compute spatial predictions for Wolfcamp

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

```
class       : SpatRaster
dimensions  : 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
```

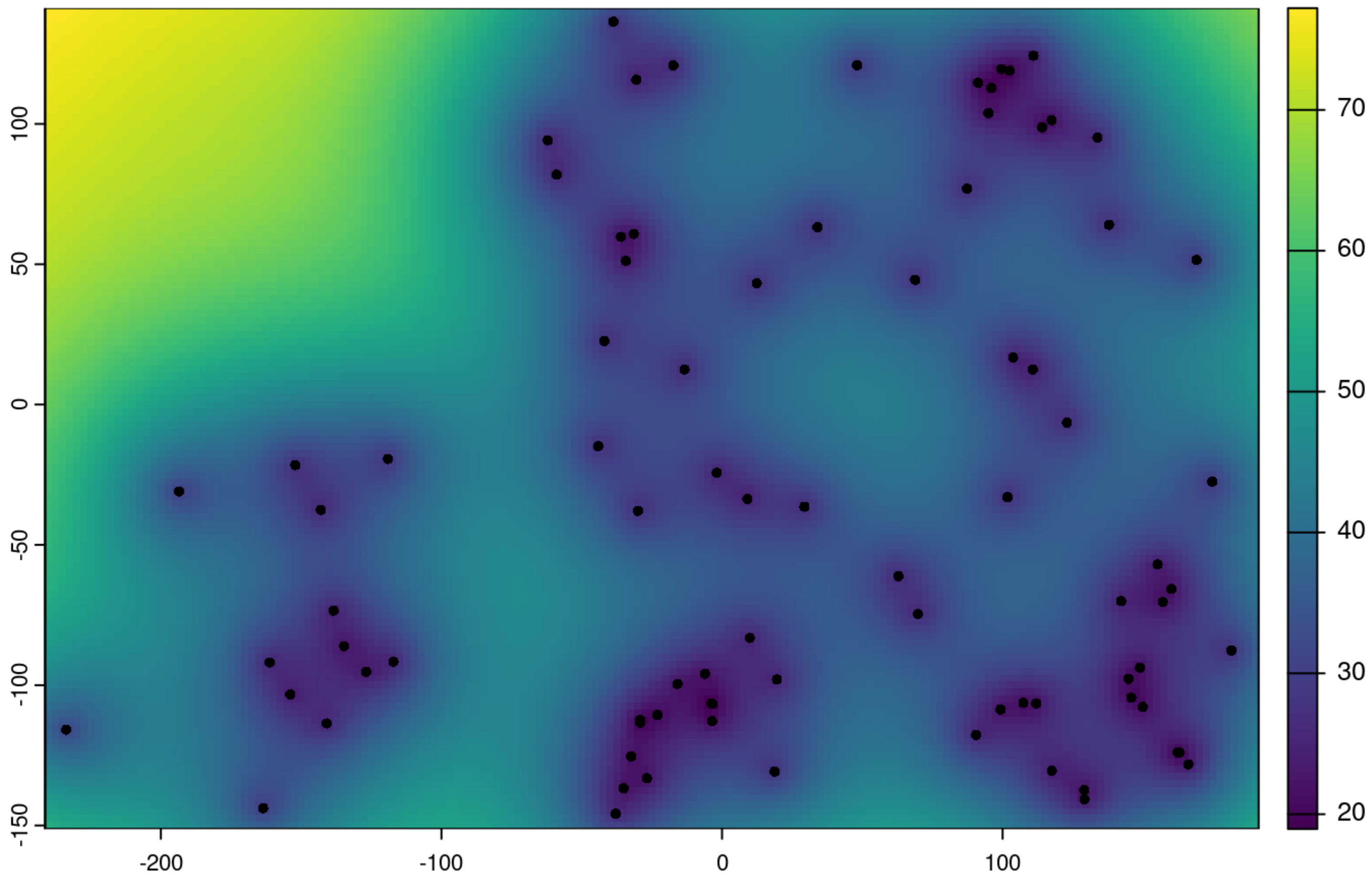
Universal kriging predictions

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



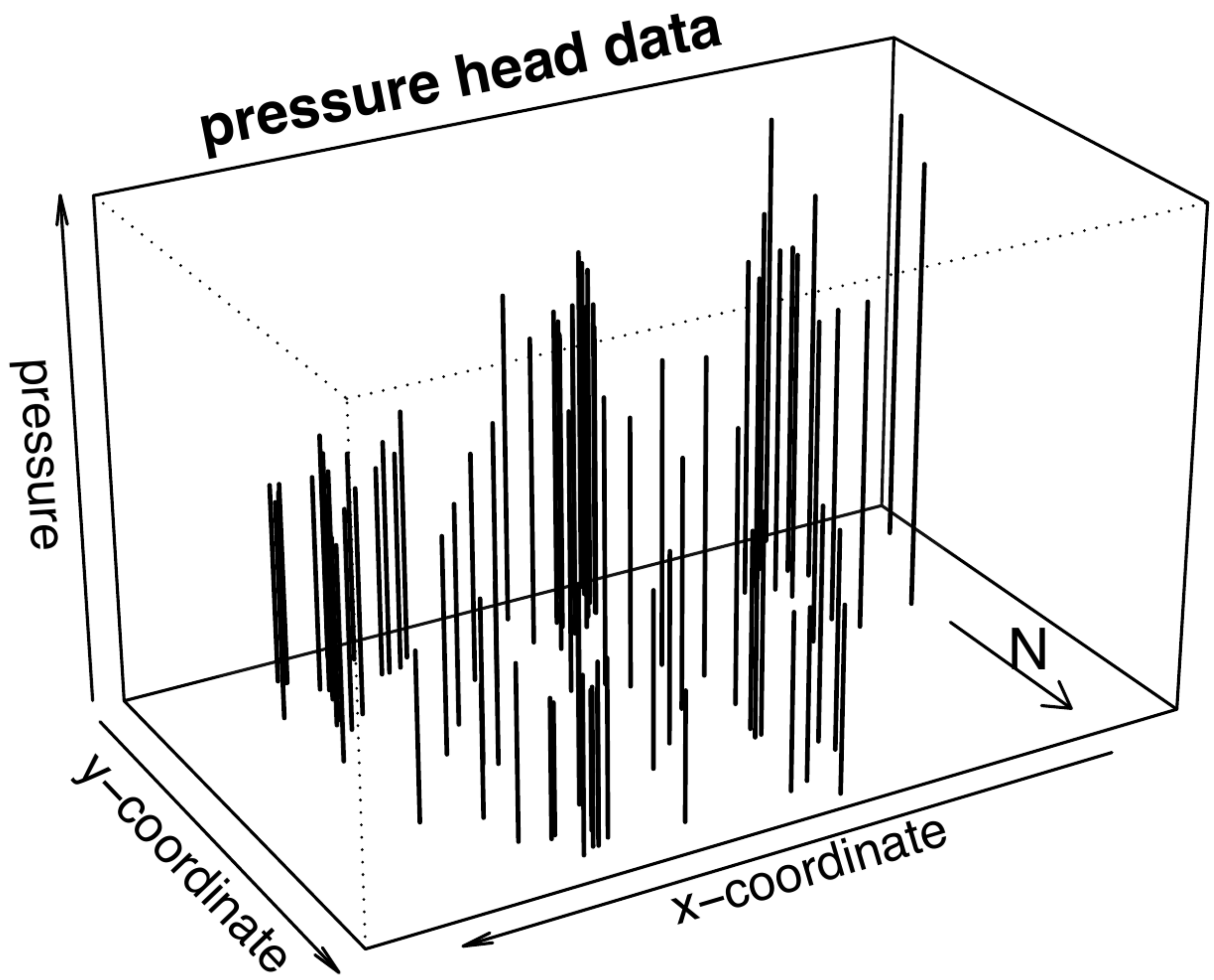
Univeral kriging standard errors

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

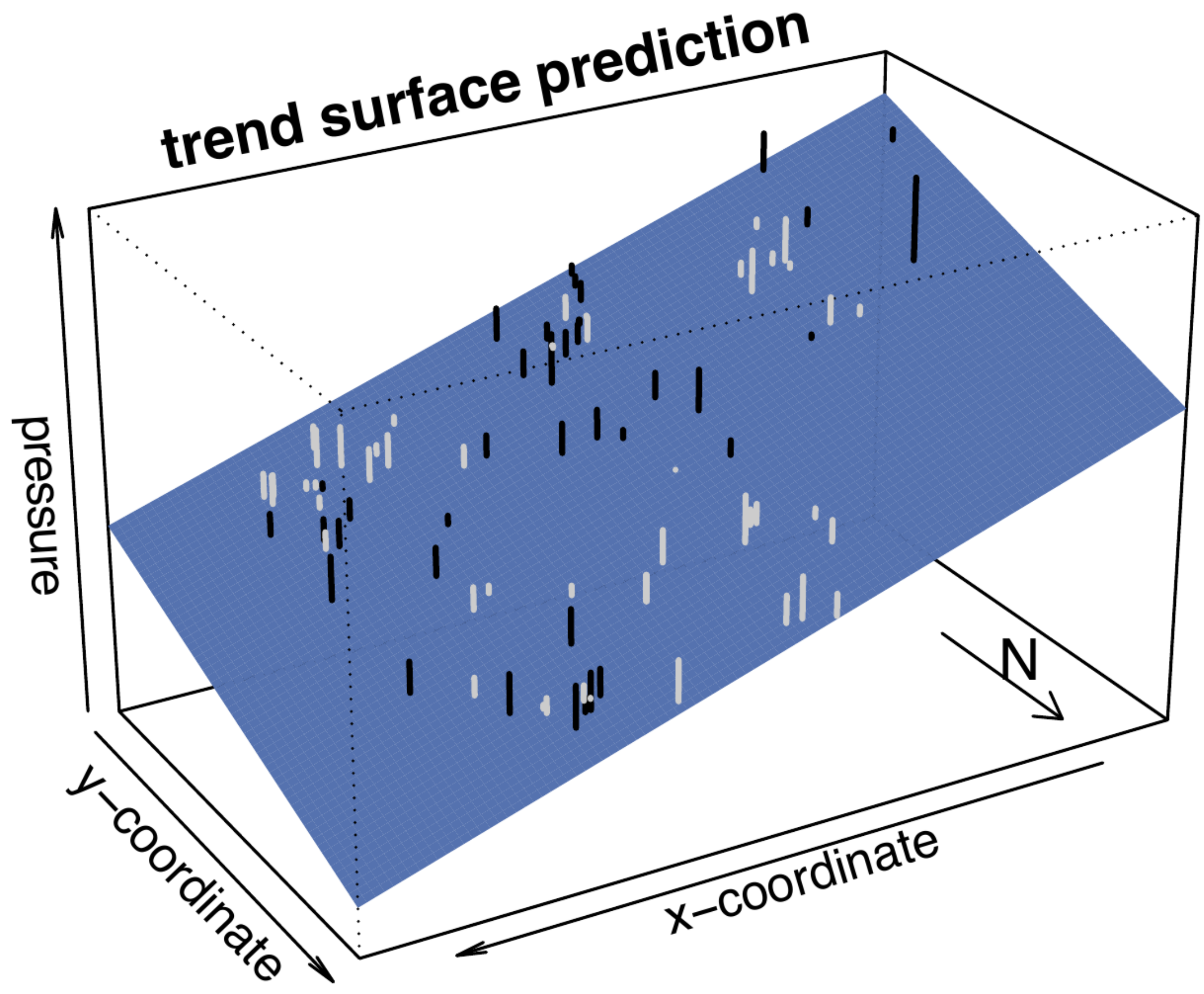


Wolfcamp aquifer data

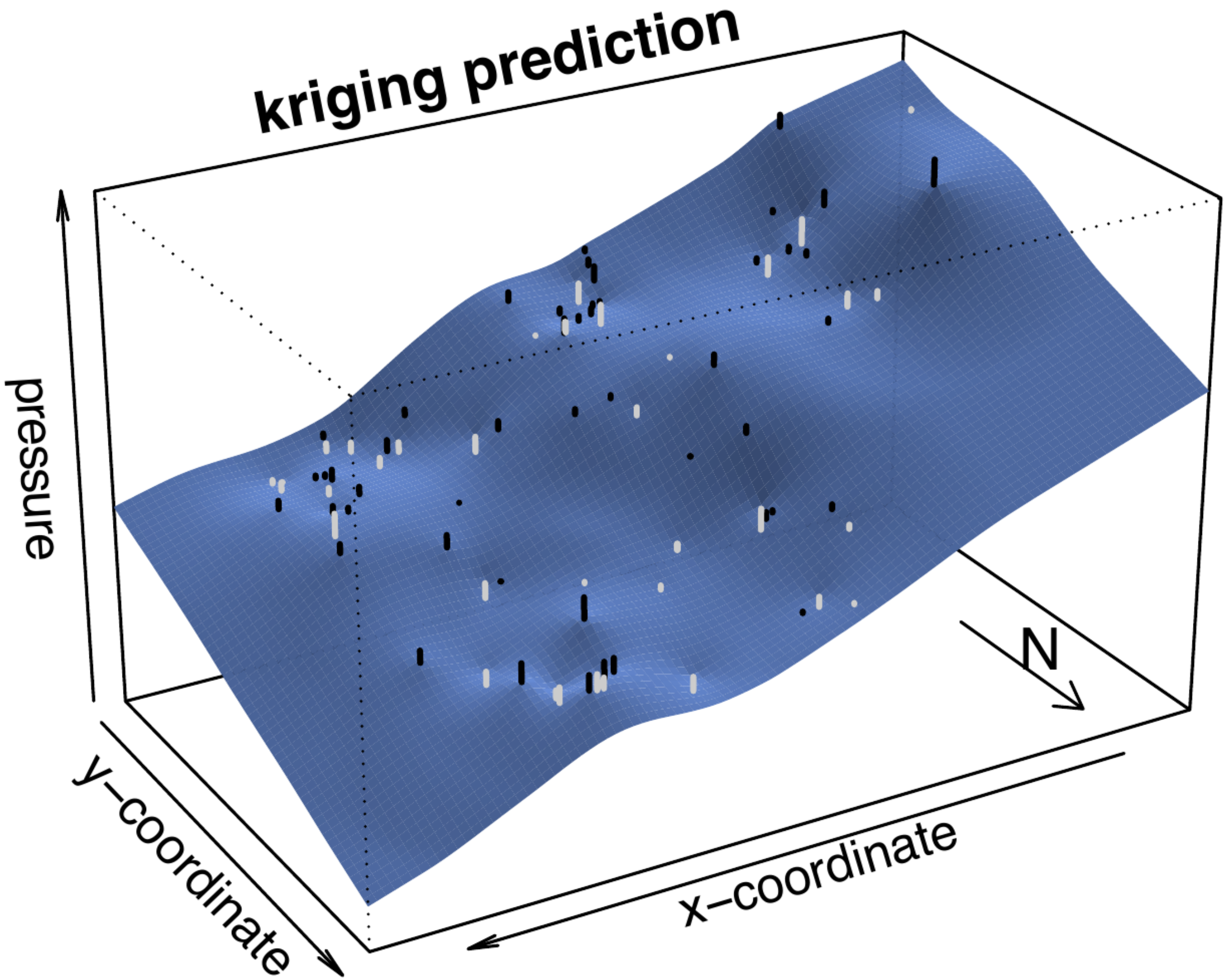
Note: 3D plot is rotated, North-East in in the lower left corner



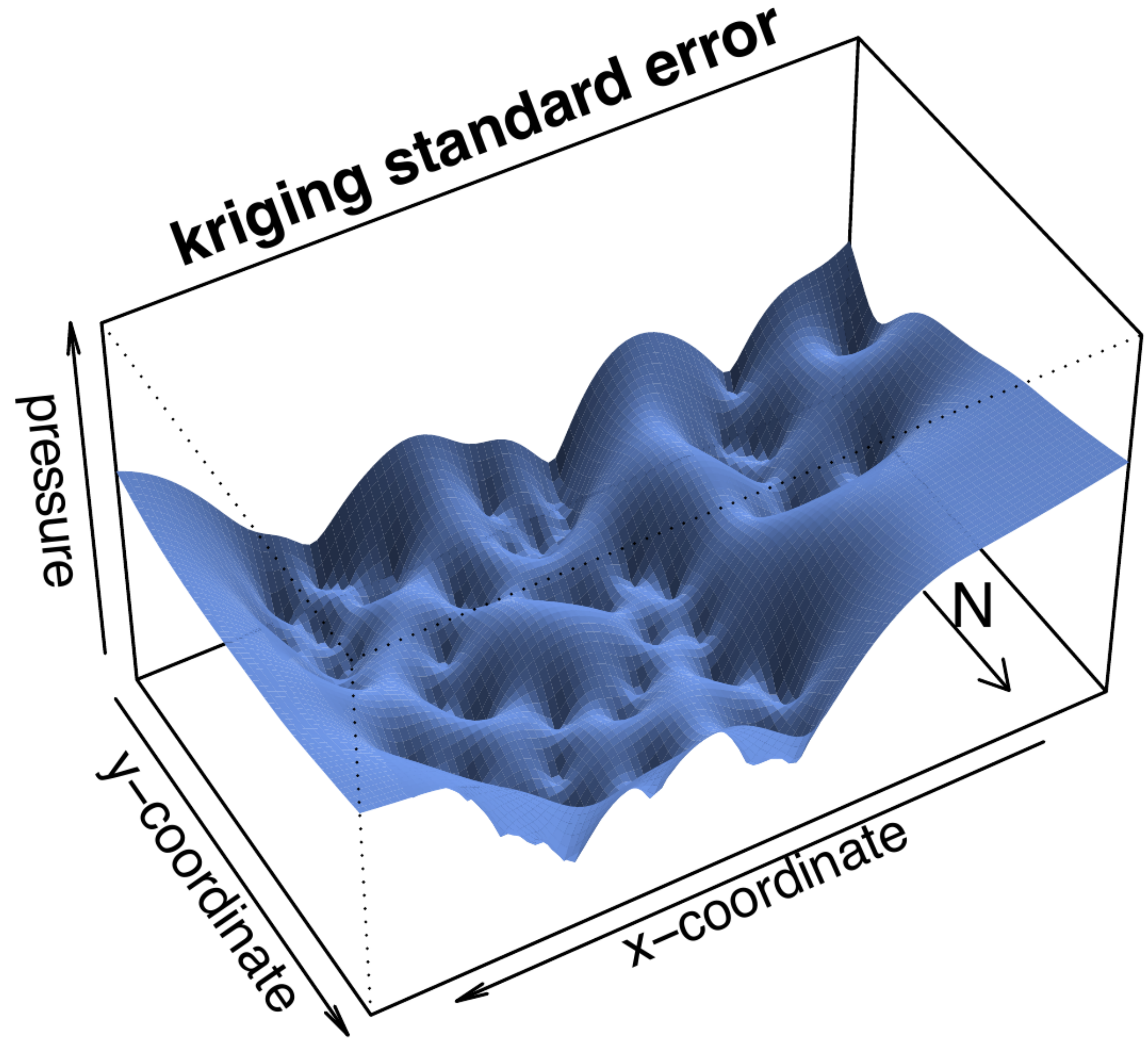
Trend prediction



Kriging prediction



Kriging standard error



3.10 Summary 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