# Spatial Statistics

Session 4

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

Andreas Papritz
(ehem. ETHZ)

21 Oct 2024

# Overview this session

- 2 special cases to expand the geostatistical model
  - lognormal kriging
  - anisotropy
- Model assessment
  - Strategies
  - Metrics
- Outlook on other techniques

# 1 Geostatistical model:
# 2 special cases

# pro memoria: Model for Gaussian spatial data

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

- with $Y_i$: $i^{\text{th}}$ datum; $S(\mathbf{x}_i)$: "signal" (true quantity) at location $\mathbf{x}_i$; $\mu(\mathbf{x}_i)$: trend
- $\{E(\mathbf{x}_i)\}$: a zero-mean Gaussian process, parametrized by covariance function $\gamma(\mathbf{h}; \theta)$ or variogram $V(\mathbf{h}; \theta)$
- $Z_i$: iid Gaussian measurement error with variance $\tau^2$

Trend $\mu(\mathbf{x}_i)$ modeled by linear regression model with spatial covariates $d_k(\mathbf{x}_i)$

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

# 1.1 Lognormal universal kriging

Gaussian model fitted to log-transformed response variable $Y(\mathbf{x}) = \log\big(U(\mathbf{x})\big)$

(e.g. often pollution datasets)

⇒ Computing UK predictions for log-transformed response

⇒ How should we back-transform to original scale of response?

Lognormal distribution

$$Y = \log(U) \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$$

Expectation and variance of $U$

$$\begin{aligned} \mathbb{E}[U] &= \mu_U = \exp(\mu_Y + 0.5\,\sigma_Y^2) \\ Var[U] &= \mu_U^2(\exp(\sigma_Y^2) - 1) \end{aligned}$$

# Backtransformation for lognormal universal kriging

$\exp\left(\hat{S}_{\mathrm{k}}(\mathbf{x}')\right)$ is a biased predictor of $U(\mathbf{x}')$

Unbiased back-transformation

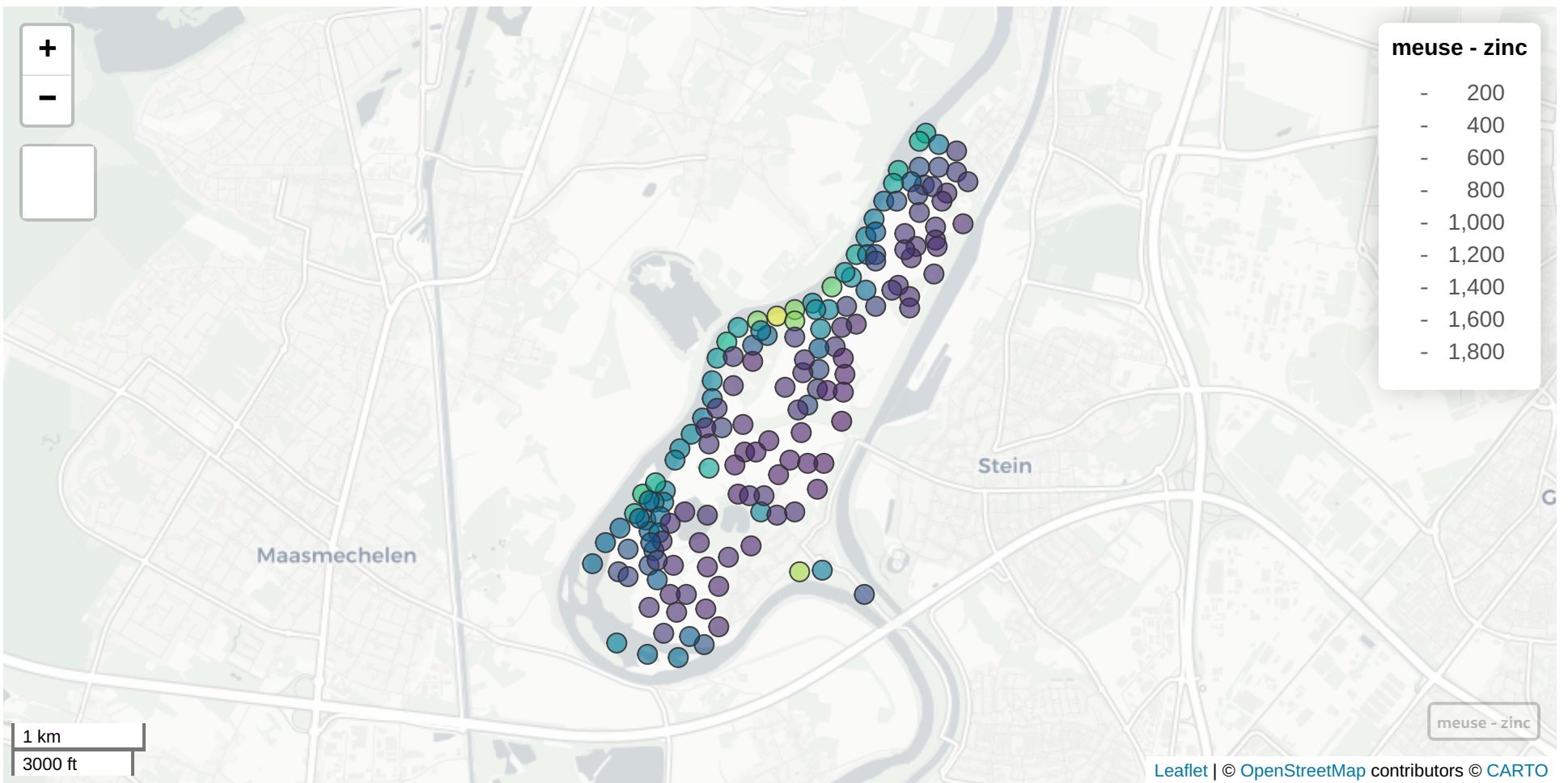
$$\hat{U}_{\mathrm{lk}}(\mathbf{x}') = \exp\left(\hat{S}_{\mathrm{k}}(\mathbf{x}') + 0.5\left\{Var[S(\mathbf{x}')] - Var[\hat{S}_{\mathrm{k}}(\mathbf{x}')]\right\}\right)$$

Limits of prediction intervals can be back-transformed directly by $\exp()$.

Back-transformation implemented in function `lgnpp` of R package `georob`.

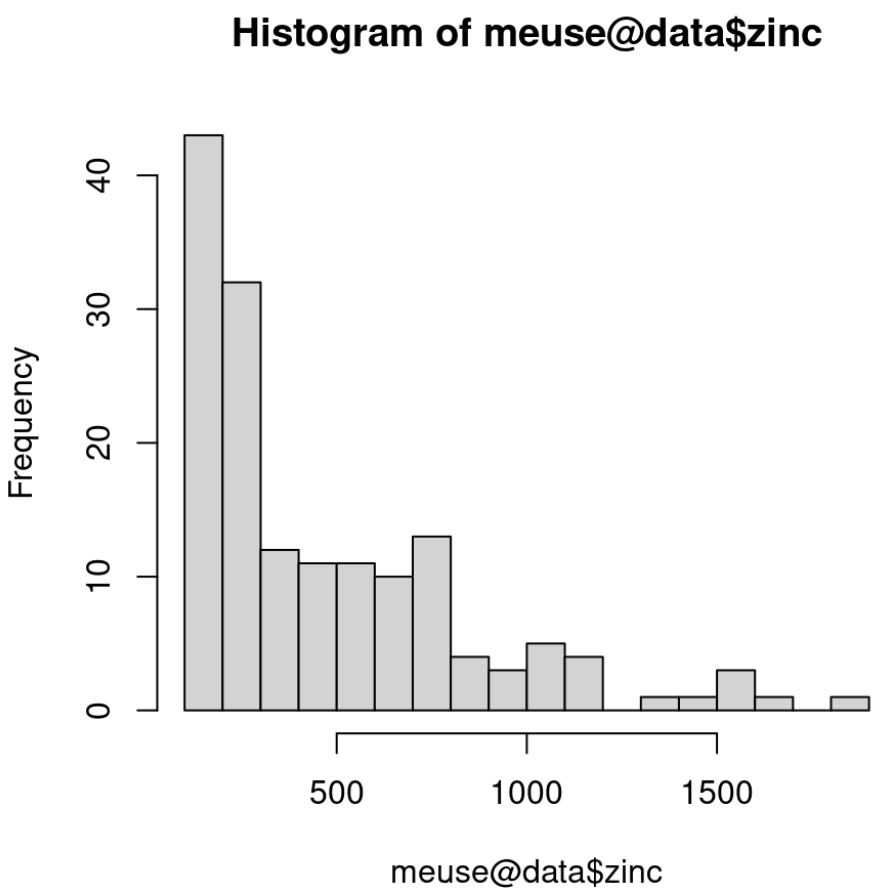
# Example: Lognormal UK Meuse `zinc` data

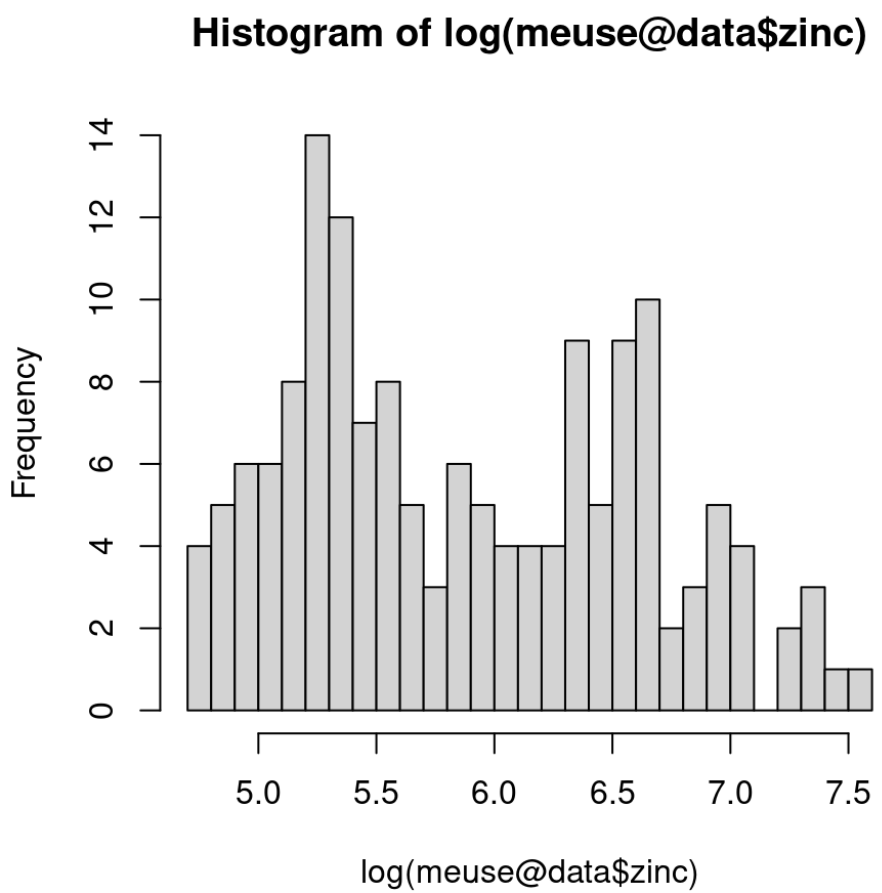
```
1  library(georob)
2  library(mapview)
3  data(meuse)
4  coordinates(meuse) <- ~x+y
5  proj4string(meuse) <- CRS("+init=epsg:28992")
6  mapview(meuse, zcol = "zinc")
```



# Example Meuse zn: Response distribution

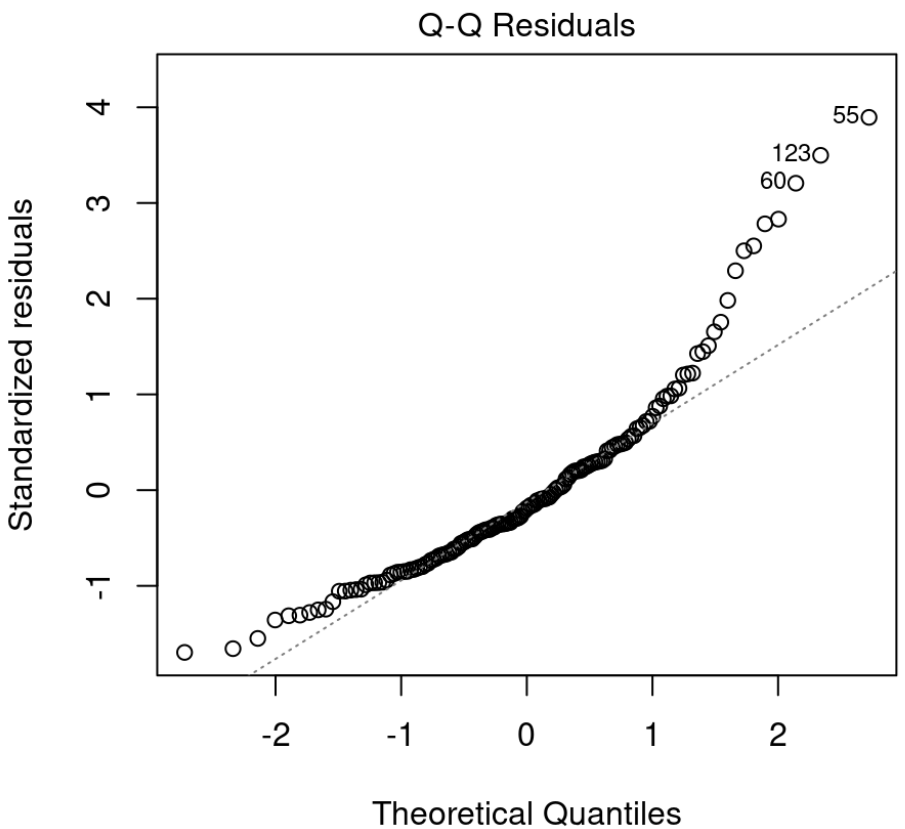
```
1 par(mfrow=c(1,2))
2 hist(meuse@data$zinc, breaks = 20)
3 hist(log(meuse@data$zinc), breaks = 20)
```

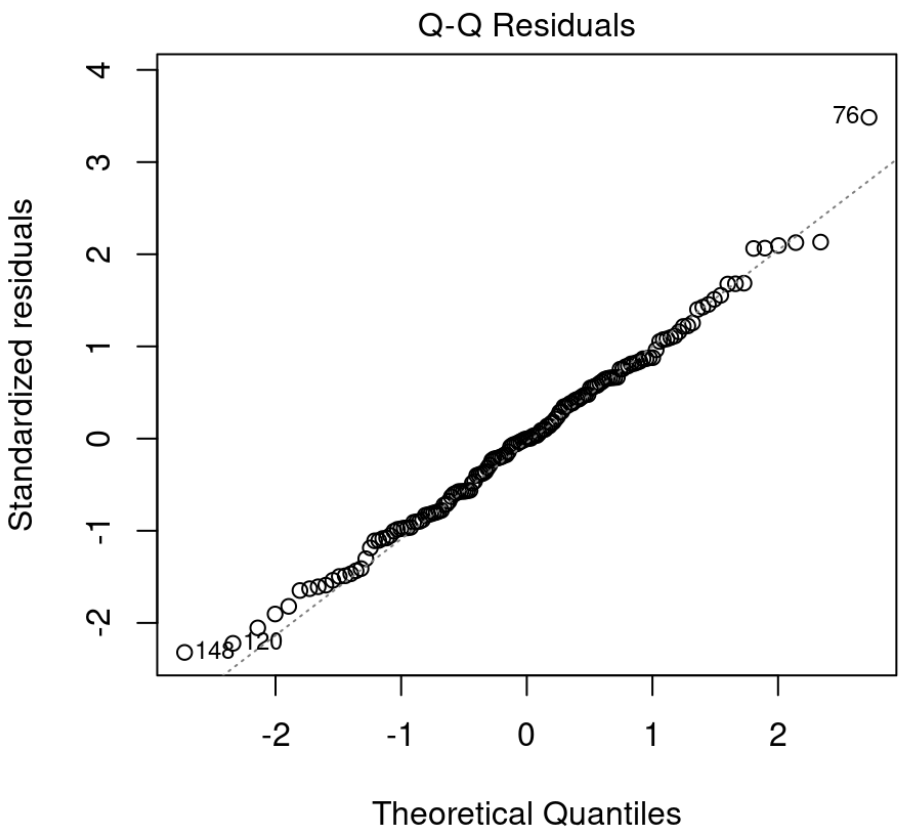


# Example Meuse zn: OLS residuals

```
1 par(mfrow = c(1,2))
2 plot(lm(zinc ~ dist, meuse), 2)
3 plot(lm(log(zinc) ~ dist, meuse), 2)
```



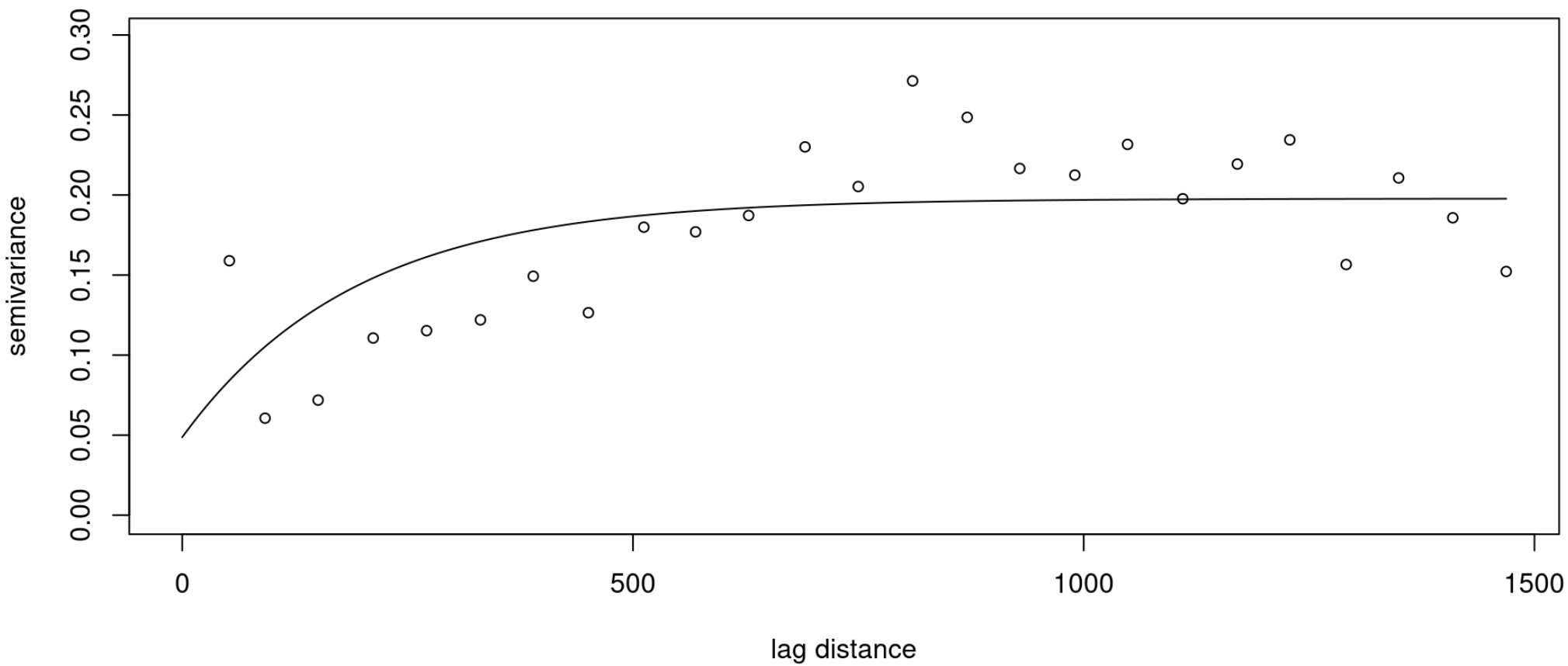
# Example Meuse zn: REML fit

```
1  r.logzn <- georob(log(zinc)~sqrt(dist), meuse, locations=~x+y,
2                    variogram.model="RMexp",
3                    param=c(variance=0.15, nugget=0.05, scale=200),
4                    tuning.psi=1000, control=control.georob(
5                      cov.bhat=TRUE, cov.bhat.betahat=TRUE))
6  r.logzn
```

```
Tuning constant:  1000

Fixed effects coefficients:
(Intercept)   sqrt(dist)
      6.985       -2.567

Variogram:  RMexp
        variance    snugget(fixed)            nugget             scale
         0.14910           0.00000           0.04867         192.52854
```

# Example Meuse zn: REML variogram

```
1 plot(r.logzn, lag.dist.def=60, max.lag=1500)
```



# Example Meuse zn: Universal kriging prediction

```
1  data(meuse.grid)
2  coordinates(meuse.grid) <- ~x+y
3  gridded(meuse.grid) <- TRUE
4  r.luk <- predict(r.logzn, newdata=meuse.grid,
5    control=control.predict.georob(extended.output=TRUE))
6  str(r.luk@data)
```

```
'data.frame':   3103 obs. of  8 variables:
 $ pred           : num  7.03 7.05 6.75 6.49 7.07 ...
 $ se             : num  0.362 0.335 0.342 0.349 0.288 ...
 $ lower          : num  6.32 6.39 6.08 5.8 6.5 ...
 $ upper          : num  7.73 7.7 7.42 7.17 7.63 ...
 $ trend          : num  6.99 6.99 6.7 6.45 6.99 ...
 $ var.pred       : num  0.0388 0.0539 0.0437 0.0357 0.0787 ...
 $ cov.pred.target: num  0.0285 0.0453 0.0379 0.0315 0.0724 ...
 $ var.target     : num  0.149 0.149 0.149 0.149 0.149 ...
 - attr(*, "variogram.object")=List of 1
  ..$ :List of 9
  .. ..$ variogram.model: chr "RMexp"
  .. ..$ param          : Named num [1:4] 0.1491 0 0.0487 192.5285
  .. .. ..- attr(*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"
  .. ..$ fit.param      : Named logi [1:4] TRUE FALSE TRUE TRUE
  .. .. ..- attr(*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"
  .. ..$ isotropic      : logi TRUE
  .. ..$ aniso          : Named num [1:5] 1 1 90 90 0
  .. .. ..- attr(*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...
  .. ..$ fit.aniso      : Named logi [1:5] FALSE FALSE FALSE FALSE FALSE
  .. .. ..- attr(*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...
  .. ..$ sincos         :List of 6
  .. .. ..$ co: num 6.12e-17
  .. .. ..$ so: num 1
```

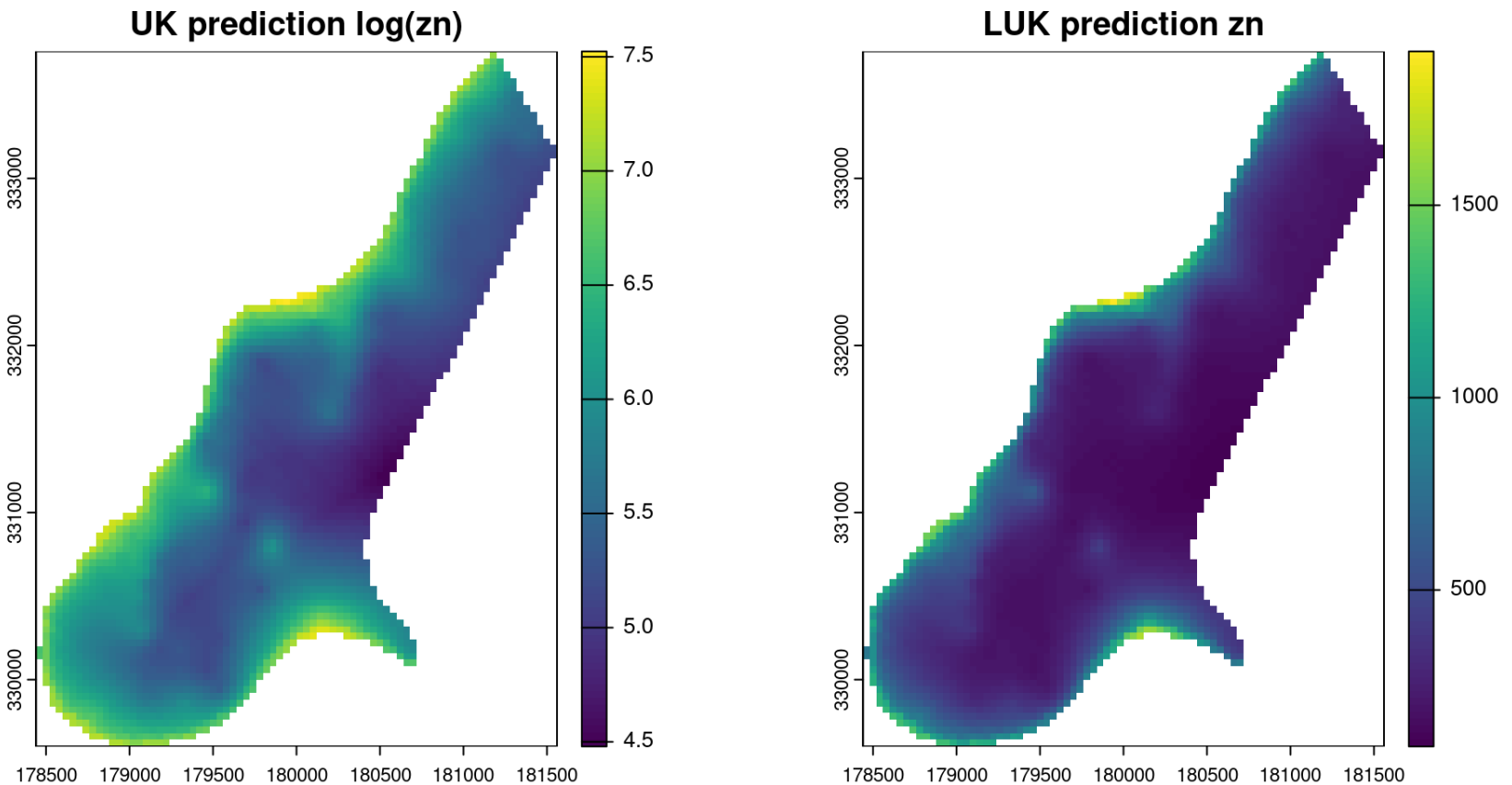
# Example Meuse zn: Unbiased backtransformation

```
1  r.luk <- lgnpp(r.luk)
2  str(r.luk@data)
```

```
'data.frame':   3103 obs. of  12 variables:
 $ pred           : num  7.03 7.05 6.75 6.49 7.07 ...
 $ se             : num  0.362 0.335 0.342 0.349 0.288 ...
 $ lower          : num  6.32 6.39 6.08 5.8 6.5 ...
 $ upper          : num  7.73 7.7 7.42 7.17 7.63 ...
 $ trend          : num  6.99 6.99 6.7 6.45 6.99 ...
 $ var.pred       : num  0.0388 0.0539 0.0437 0.0357 0.0787 ...
 $ cov.pred.target: num  0.0285 0.0453 0.0379 0.0315 0.0724 ...
 $ var.target     : num  0.149 0.149 0.149 0.149 0.149 ...
 $ lgn.pred       : num  1189 1204 902 694 1215 ...
 $ lgn.se         : num  440 409 314 249 354 ...
 $ lgn.lower      : num  554 595 438 331 667 ...
 $ lgn.upper      : num  2286 2215 1673 1299 2064 ...
 - attr(*, "variogram.object")=List of 1
  ..$ :List of 9
  .. ..$ variogram.model: chr "RMexp"
  .. ..$ param          : Named num [1:4] 0.1491 0 0.0487 192.5285
  .. .. ..- attr(*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"
  .. ..$ fit.param      : Named logi [1:4] TRUE FALSE TRUE TRUE
  .. .. ..- attr(*, "names")= chr [1:4] "variance" "snugget" "nugget" "scale"
  .. ..$ isotropic      : logi TRUE
  .. ..$ aniso          : Named num [1:5] 1 1 90 90 0
  .. .. ..- attr(*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...
  .. ..$ fit.aniso      : Named logi [1:5] FALSE FALSE FALSE FALSE FALSE
  .. .. ..- attr(*, "names")= chr [1:5] "f1" "f2" "omega" "phi" ...
```

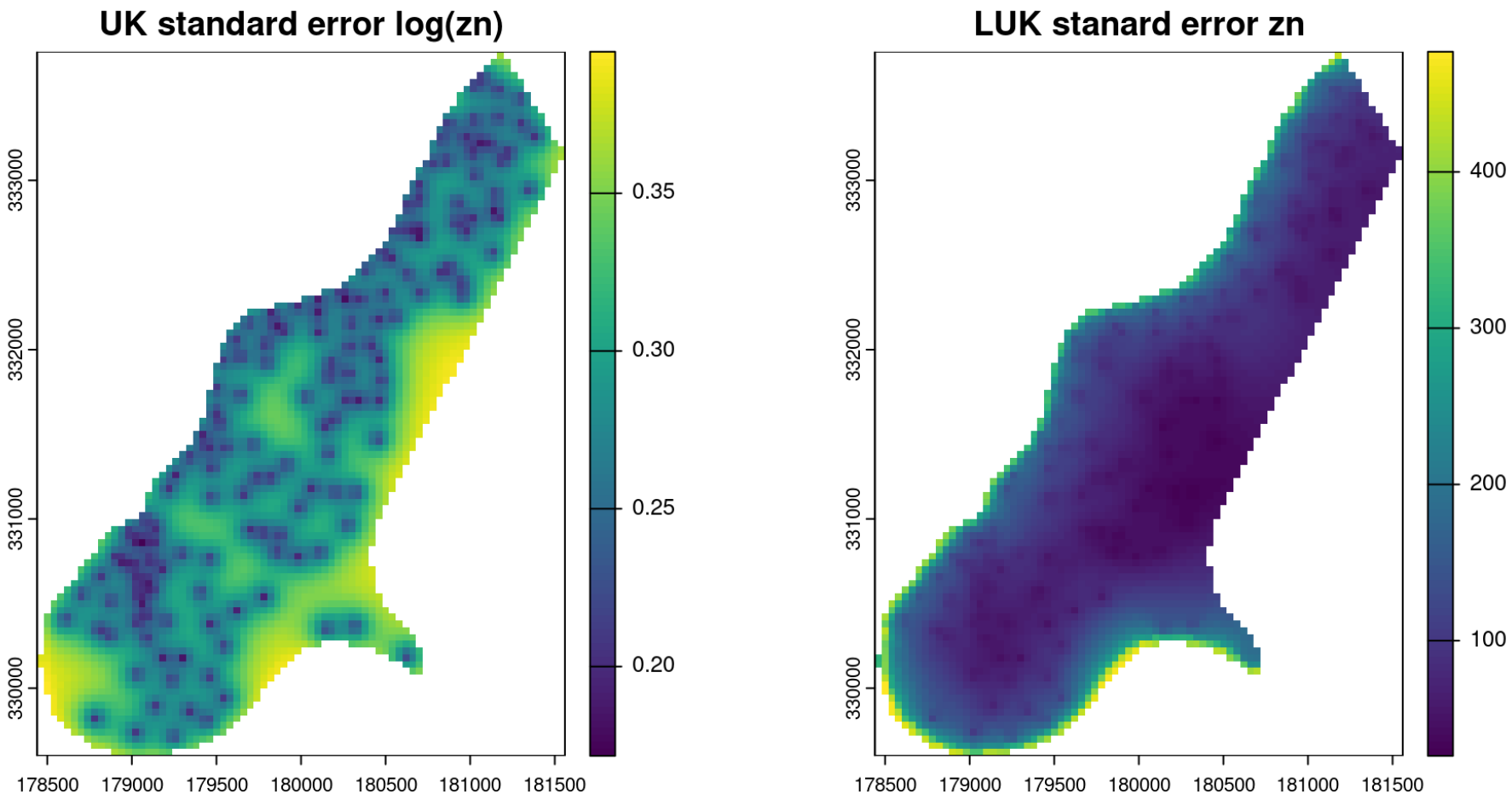
# Example Meuse zn: Predicted maps

```
1  library(terra)
2  r.luk <- rast(r.luk)
3  par(mfrow=c(1,2))
4  plot(r.luk["^pred$"], main = "UK prediction log(zn)")
5  plot(r.luk["lgn.pred"], main = "LUK prediction zn")
```



# Example Meuse zn: Predicted standard errors

```
1  par(mfrow=c(1,2))
2  plot(r.luk["^se$"], main = "UK standard error log(zn)")
3  plot(r.luk["lgn.se"], main = "LUK stanard error zn")
```



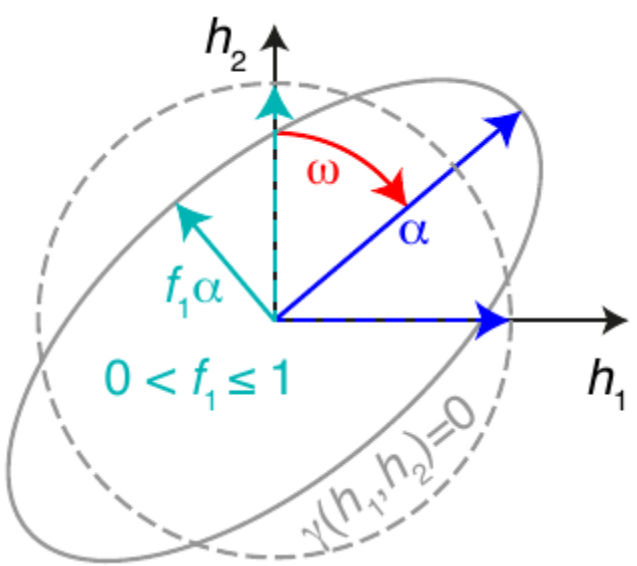
# 1.2 Anisotropic variograms

Spatial auto-correlation can vary from one direction to the other, i.e. it can be anisotropic.

In many instances the anisotropy is such that it could be made isotropic by a simple linear transformation of the spatial coordinates. ⇒ geometrically anisotropic variogram function

Idea: rotate and stretch/shrink components of $\mathbf{x}$ such that the stochastic process is isotropic in the transformed coordinate system

$$V(h^*) = V\left(\sqrt{(\mathbf{hA})^{\mathrm{T}}\mathbf{Ah}}\right)$$

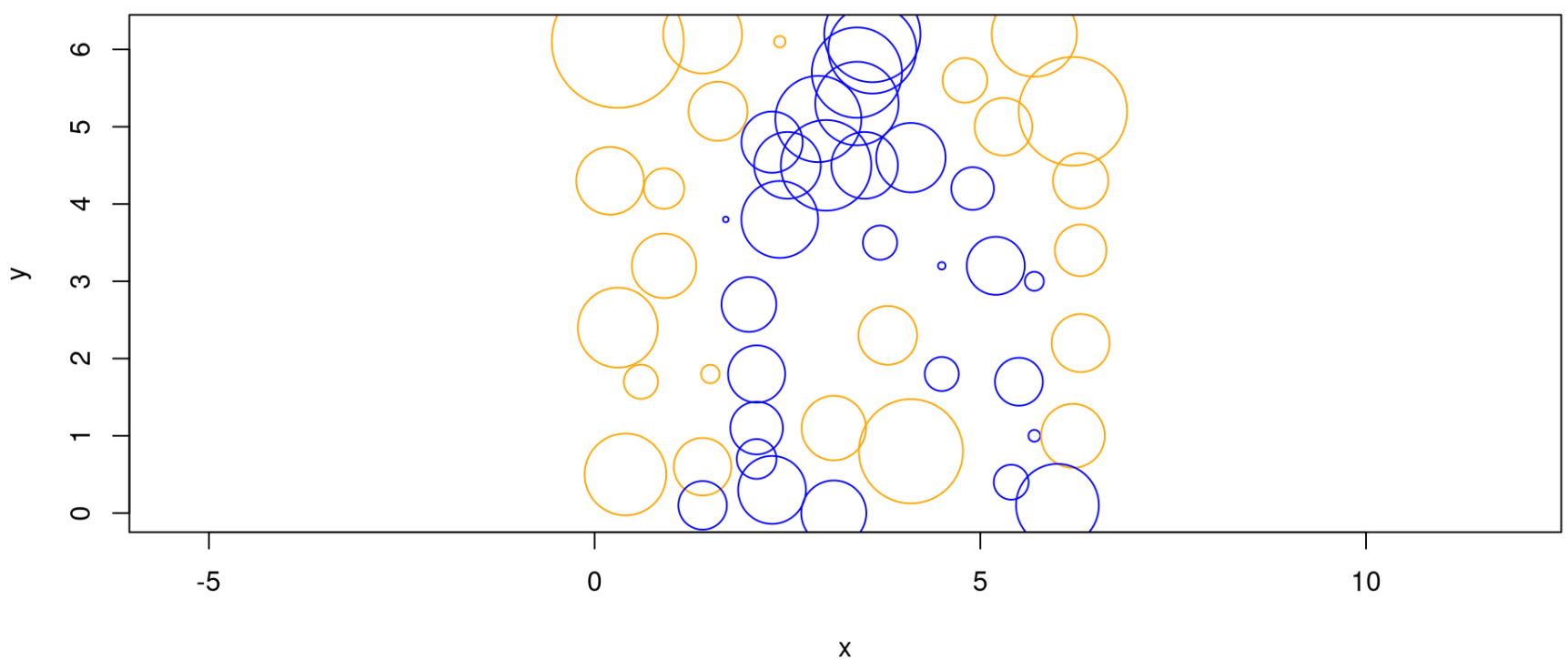



Iso-semivariances contours are ellipsoids in space of non-transformed coordinates and are mapped by transformation:

$$\mathbf{A} = \begin{bmatrix} 1/\alpha & 0 \\ 0 & 1/(f_1\alpha) \end{bmatrix} \begin{bmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{bmatrix}$$

# Example: elevation data

```
data(elevation, package="georob")
d.elevation <- elevation
d.elevation$res <- residuals(lm(height~y, d.elevation))
plot(y~x, d.elevation, cex=sqrt(abs(res)), asp=1,
  col=c("blue", NA, "orange")[sign(res)+2])
```



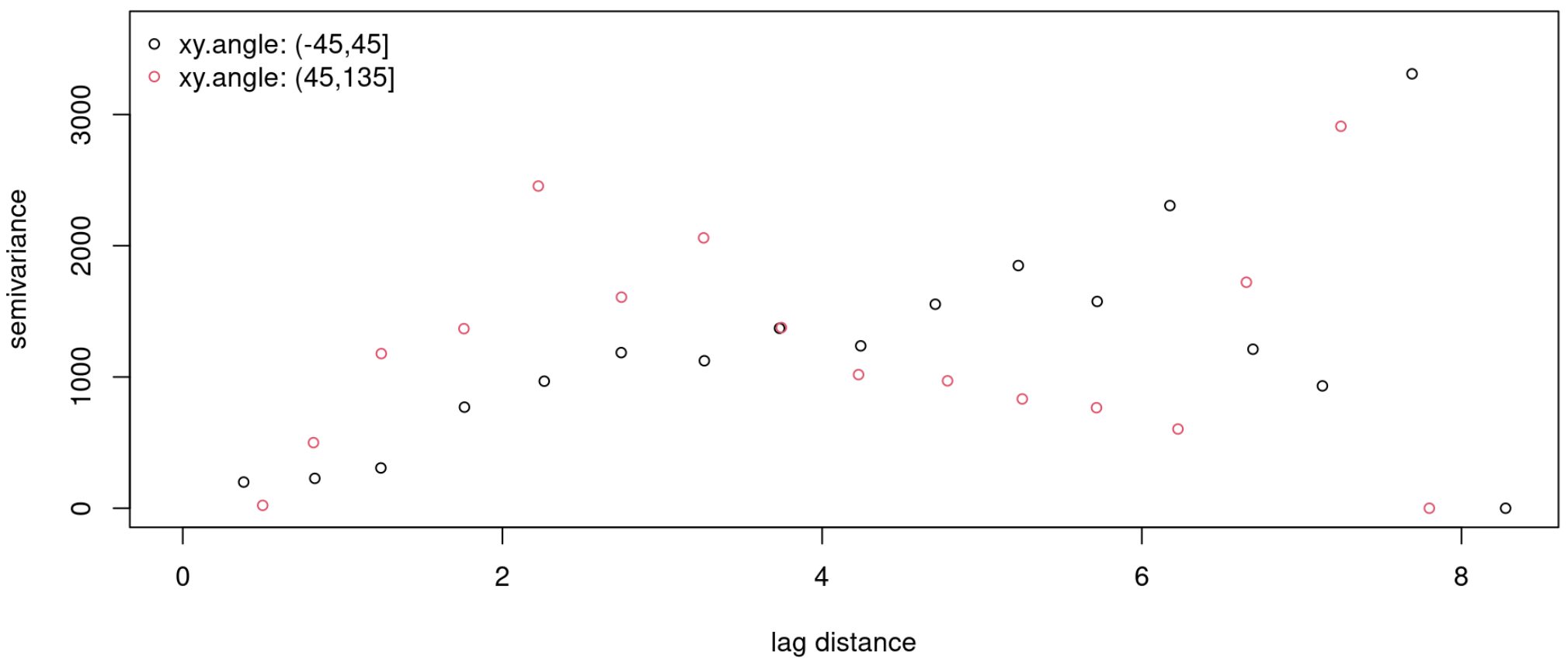
# Example: elevation data - directional sample variogram

```
1 r.sv <- sample.variogram(d.elevation$res,
2   locations=as.matrix(d.elevation[, c("x","y")]),
3   lag.dist.def=0.5, xy.angle.def=c(0, 45, 135, 180))
4 summary(r.sv)
```

```
Sample variogram estimator: qn

Summary of lag distances
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.3812  2.2244  4.2278  4.1295  6.1756  8.2759

Summary of number of pairs per lag and distance classes
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   11.00   45.00   40.18   63.00   73.00

Angle classes in xy-plane: (-45,45] (45,135]
Angle classes in xz-plane: [0,180]
```

# Example: elevation data - directional sample variogram

```
1  plot(r.sv)
```



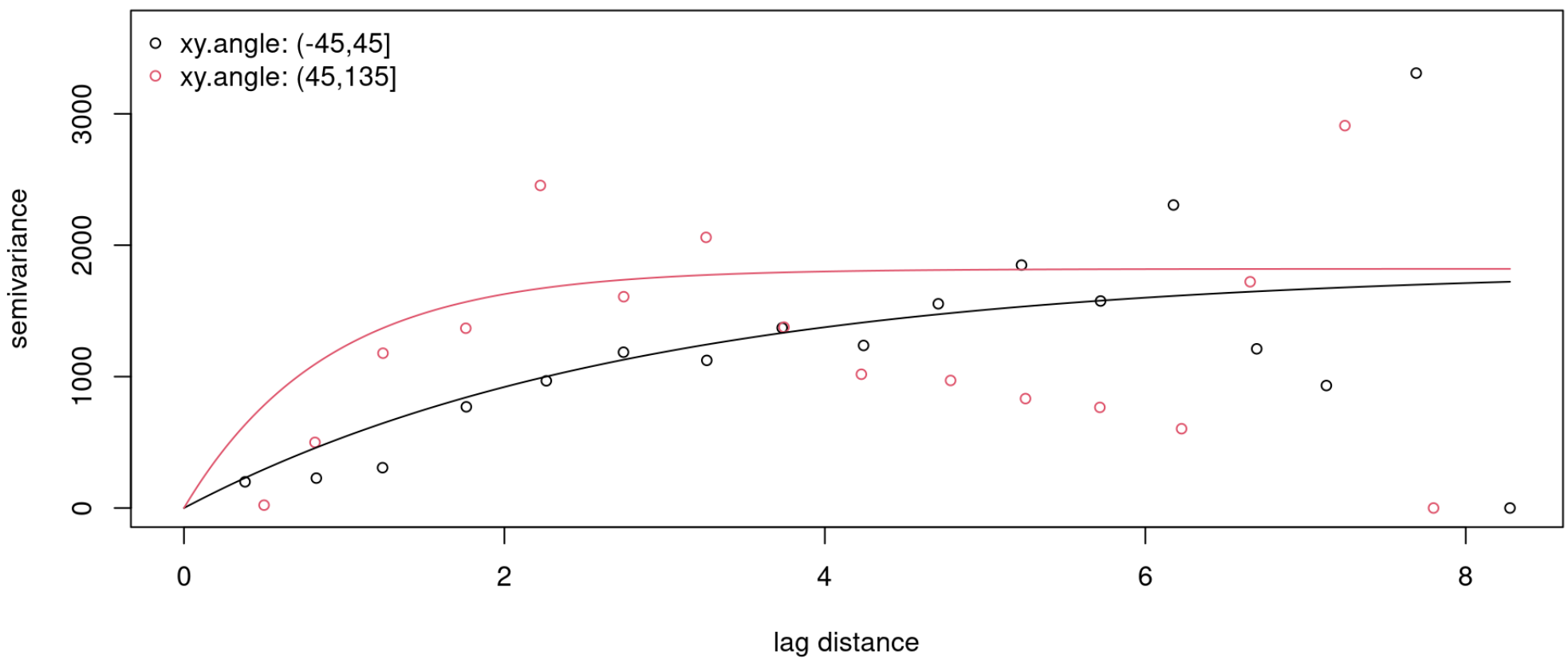
# Example: elevation data - fitted directional variogram

```
1 r.exp <- fit.variogram.model(r.sv,
2   variogram.model="RMexp",
3   param=c(variance=1500, nugget=10, scale=4),
4   aniso = default.aniso(f1 = 0.4, omega = 0),
5   fit.aniso = default.fit.aniso(f1 = TRUE, omega = TRUE))
6 r.exp
```

```
Variogram:  RMexp
        variance    snugget(fixed)            nugget             scale
       1.820e+03         0.000e+00         1.289e-07         1.256e+02

              f1         f2(fixed)             omega        phi(fixed)
        0.006757          1.000000         17.415945         90.000000
     zeta(fixed)
        0.000000
```

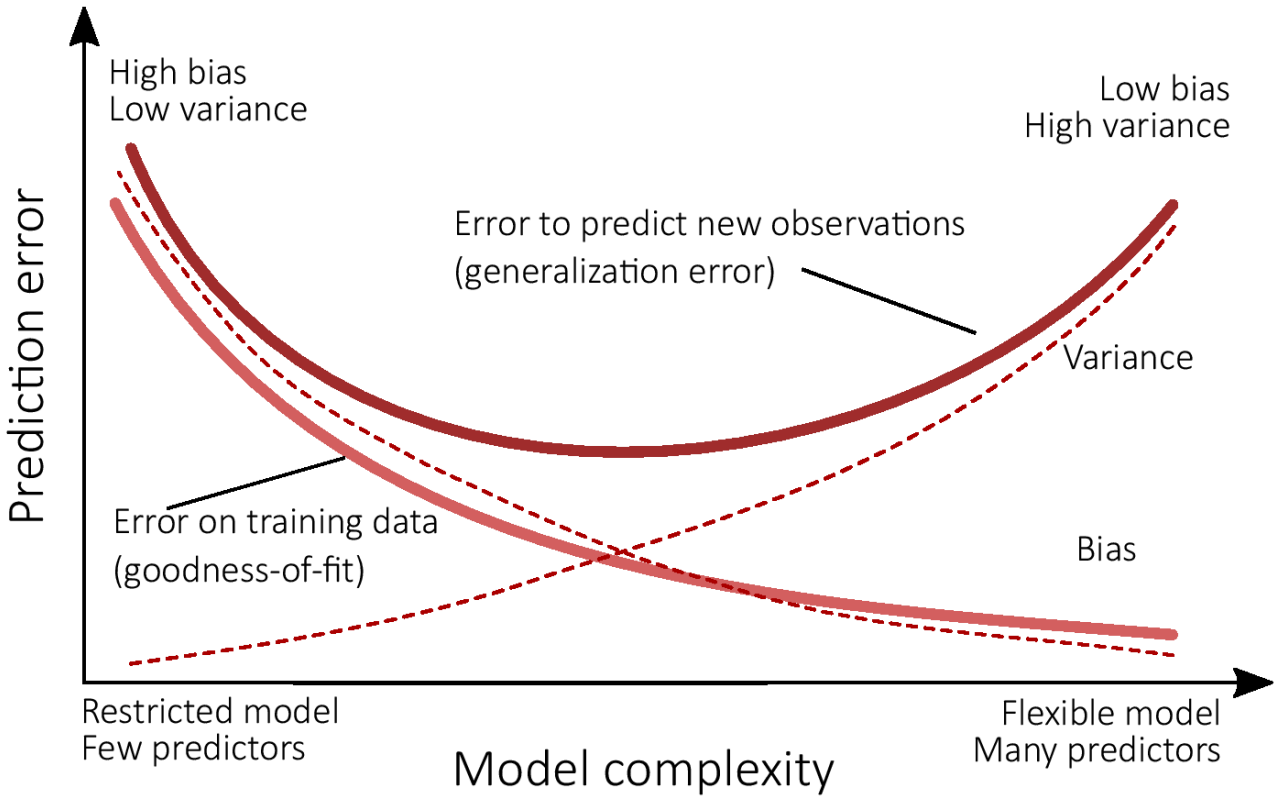
# Example: elevation data - fitted directional variogram

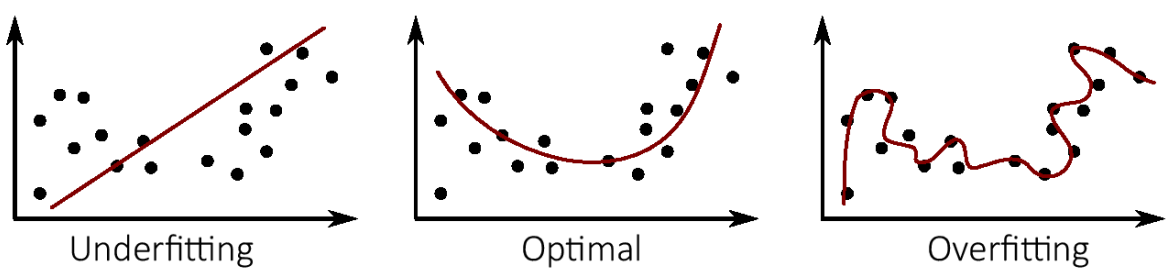
```
1 plot(r.sv)
2 lines(r.exp, xy.angle=c(0, 90))
```



# 2 Model complexity

# 2.1 Bias-Variance trade-off illustration



# Bias-Variance trade-off

Expected validation MSE for a given value $x_0$ can be decomposed into variance, squared bias and variance of the error $\epsilon$ (unexplained variation):

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

**Variance** $Var(\hat{f}(x_0))$

How much $\hat{f}$ would change if we estimated it using a different data set. High variance: small changes in the data result in large changes in $\hat{f}$.

**Bias** $Bias(\hat{f}(x_0))$

Error that is introduced by approximating a real-world problem, which may be extremely complicated, by a much simpler model. E.g. linear regression assumes linearity which is unlikely to be the true model.

# 3 Model assessment

# 3.1 Goal of model assessment

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

Summary statistics (like RMSE, $R^2$, kappa) to characterize:

- How well does a model fit the data?
- Compare models: What is the best model structure, what are relevant covariates? Model selection, variogram functions, tuning parameters.
- Evaluate generalization properties: What is the average error to predict for a location without observed data?

Models with *goodness-of-fit* metrics often allow model building, but:

- Goodness-of-fit not necessarily good criterion for judging quality of predictions.
- Covariate selection by `step()` does not lead to a unique set of covariates but depends on the search strategy and on criterion (AIC/BIC) used for model section.
- Likelihood ratio tests cannot be used to compare the goodness-of-fit of models that have different variograms.

Therefore, independent validation or data splitting are required. Machine learning models often directly rely on such techniques as *goodness-of-fit* criteria do not exist or are not meaningful.

# 3.2 Model assessment strategies

How to obtain data pairs of observed and predicted?

## *«observed/measured» vs. «predicted»*

Topsoil clay content [%]
(interval scaled)

| measured | predicted |
|---|---|
| 5.4 | 12.3 |
| 15.1 | 14.7 |
| 25.6 | 22.1 |
| 32.4 | 35.1 |
| 34.2 | 34.6 |
| ... | ... |

Rainfall occurence at specific time point [1 = yes, 0 = no]
(binary, with predicted probabilities)

| observed | Predicted probability | OR predicted class |
|---|---|---|
| 1 | 0.8 | 1 |
| 0 | 0.7 | 1 |
| 1 | 0.9 | 1 |
| 0 | 0.1 | 0 |
| ... | ... | |

Geological unit
(nominal, >2 classes, with predicted class or probability)

| observed | predicted Granite | predicted Limestone | predicted Schist |
|---|---|---|---|
| Granite | 0.2 | 0.3 | 0.5 |
| Limestone | 0.05 | 0.8 | 0.15 |
| Schist | 0.1 | 0.2 | 0.7 |
| ... | ... | | |

# Overview strategies

- Completely independent sampling
- Splitting of available data
- Cross-validation
- Non-parametric boostrap

Note: In the classical statistical context an independent data set with which the model performance is evaluated is often called **validation set**, in machine learning context it is called a **test set**. The *validation set* in machine learning is an additional subset of the data, especially often used with to neuronal networks, that serves to find optimal tuning parameters.

# Completely independent sampling

## Desired option

Independent data:

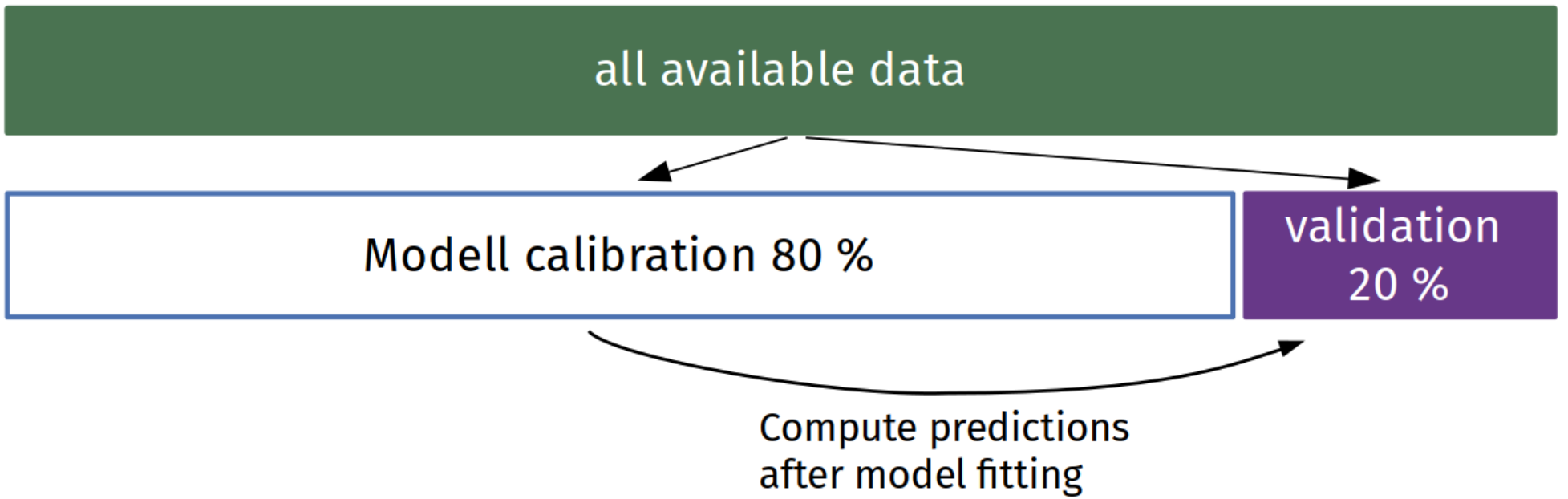
- Survey from another institution/database (but, needs to be the same target population and survey method)
- New survey:
  - Independently surveyed from data used for model calibration
  - Approach to select locations (and depth if 3D): design-based sampling, e.g. stratified random sampling.

**Problems**

- Often no funding for additional survey
- Not all studies allow for independent sampling, because it is not meaningful. e.g. landslide occurence is observed whenever it happens.

Not needed, if calibration locations were chosen by a design-based sampling approach, that allows unbiased estimate of model performance metrics, we may use data splitting.

# Data splitting

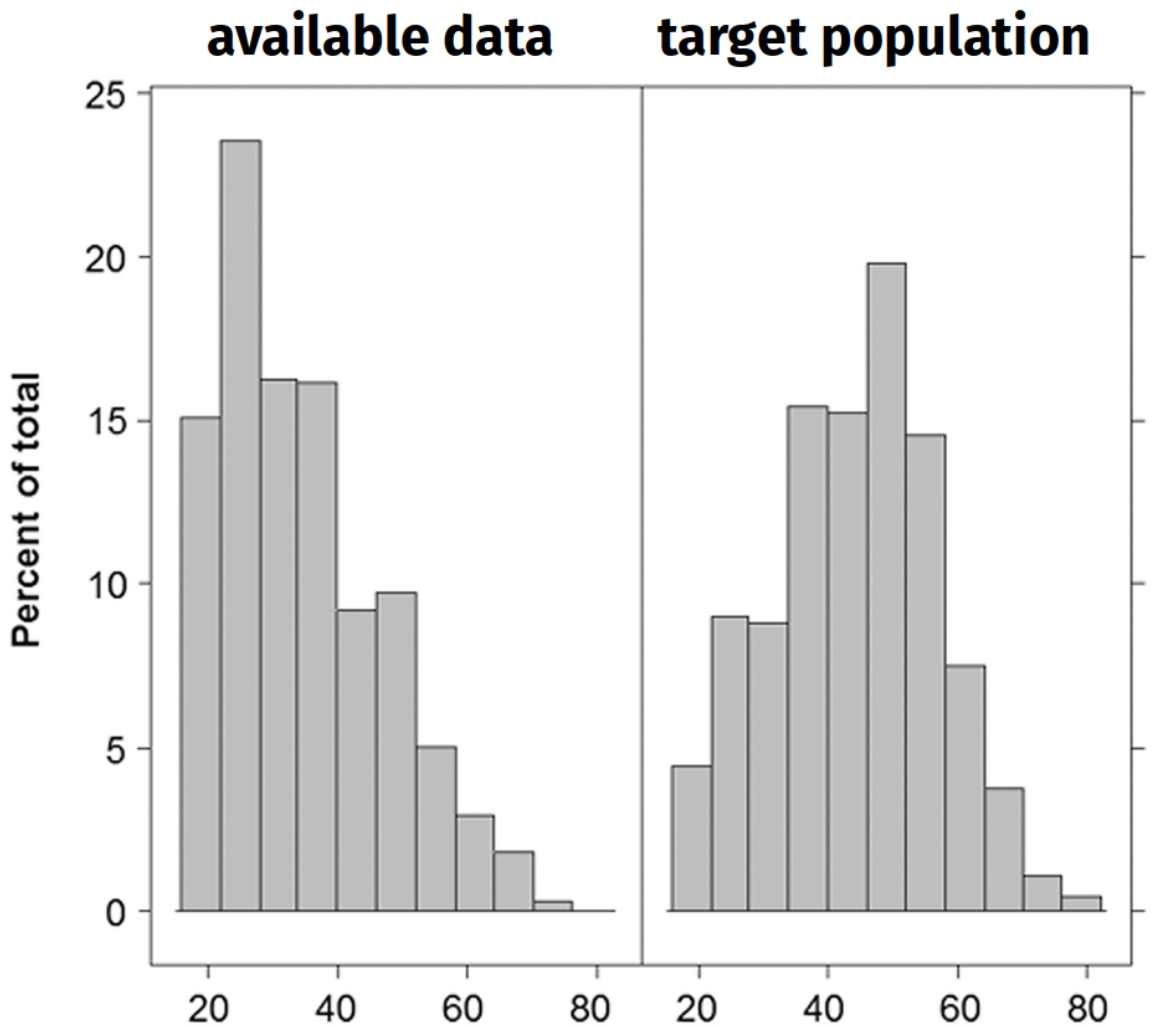



Rule of thumb: 20 % for validation, or a minimal number that allows to compute reliable metrics/means, e.g. 30–50.

How to split:

- simple random sample
- weighted or stratified random sample, if data does not represent the target population well

# Example for weighted splitting




**Weighted random sample**

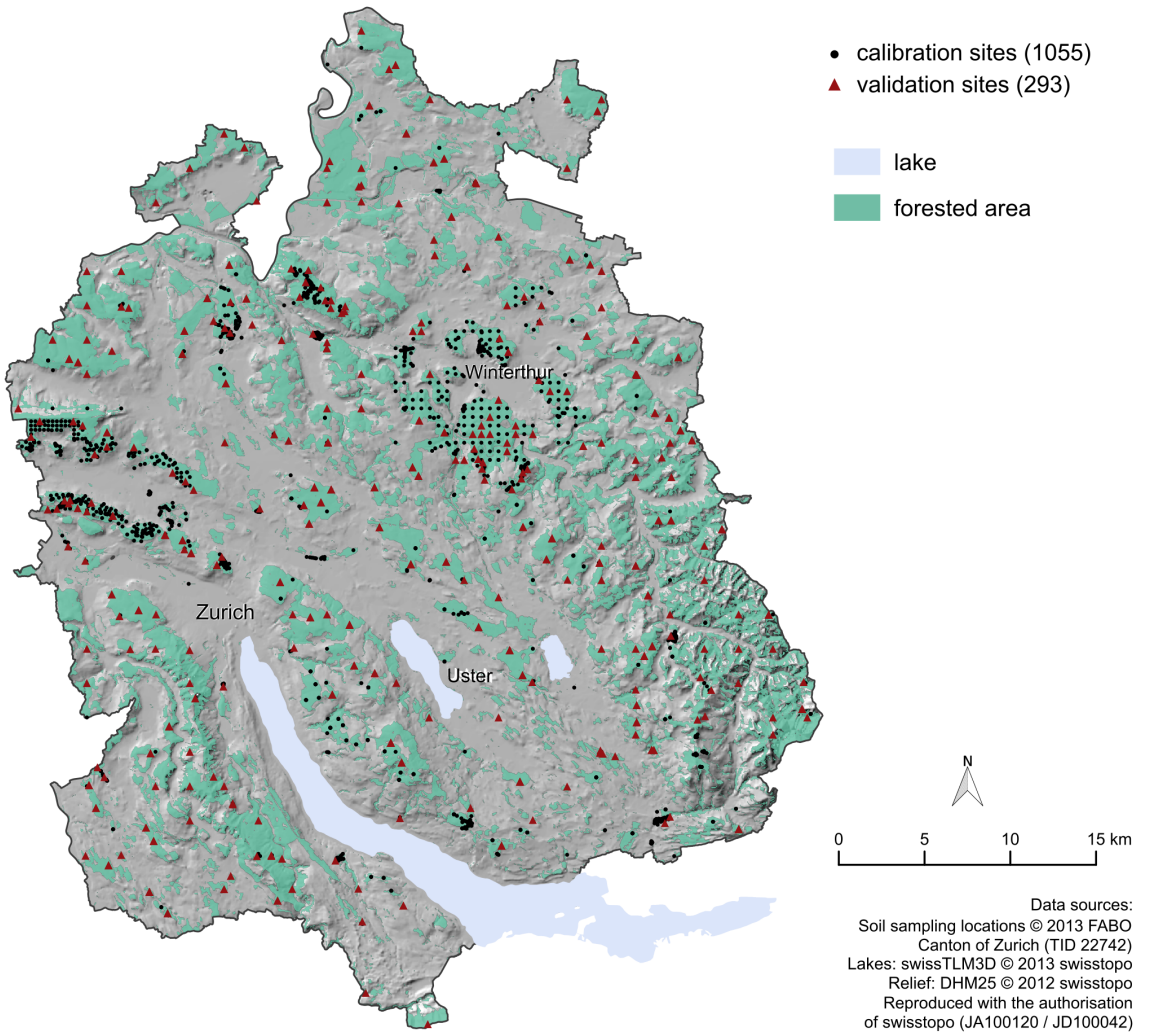
Use occurrence probabilities of target population to obtain a validation set with distribution of target population.

# Example for weighted splitting by spatial distribution

Problem: Data from pooled surveys, some strongly spatially clustered. Goal: Prediction for complete forested area of Canton of Zurich

→ Data splitting with weights to obtain homogenouous cover of total study area.

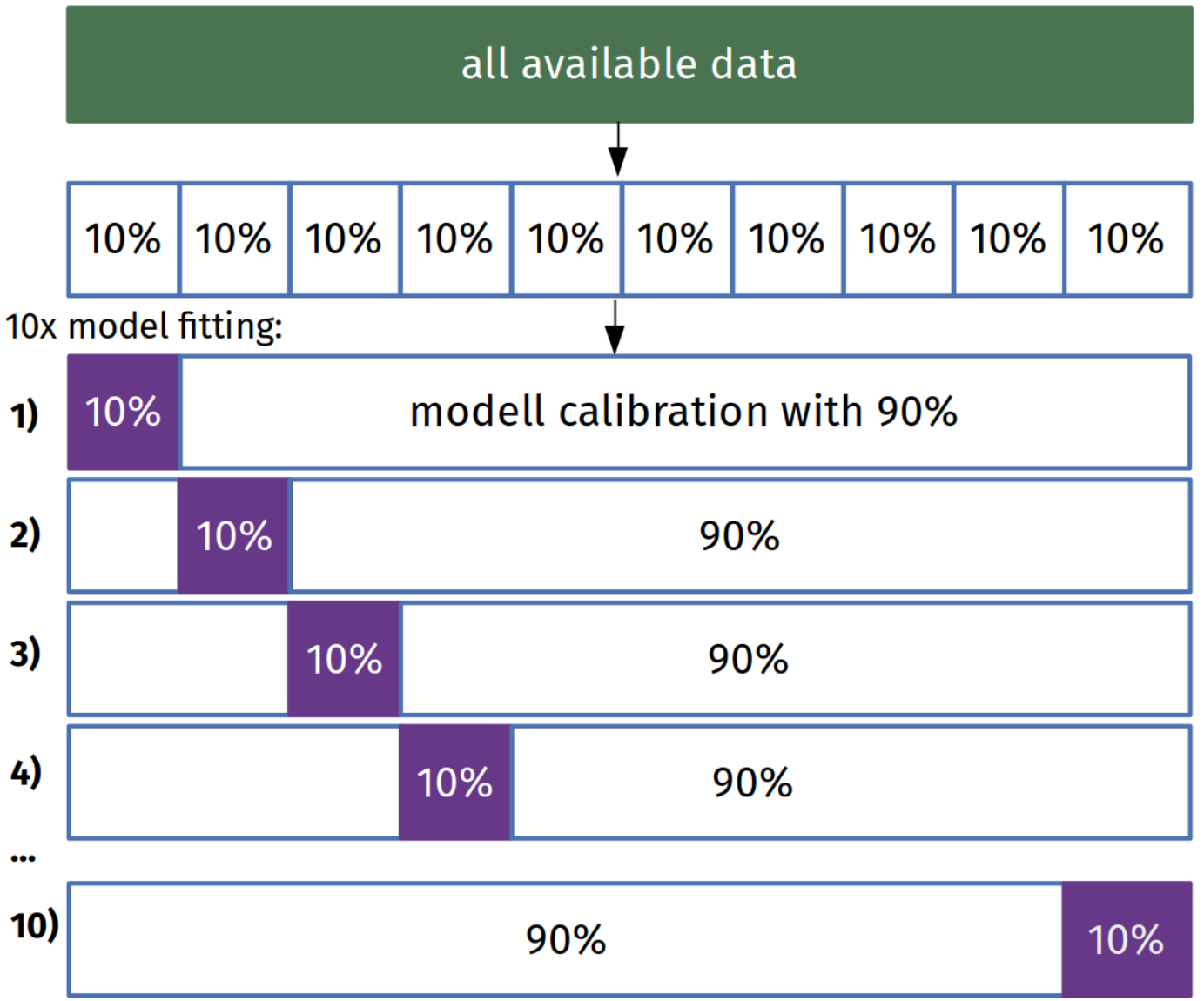
(Weights computed from Thyssen-polygons intersected with the forested area.)



# Data splitting – problems

- No proper independent data
- Sub-optimal variant of new survey
- Largely depends how splitting is done
- Especially if splitting is done randomly: better to repeat

# k-fold cross-validation




*k = 10*
Splitting: randomly

10% are not used for model building. In each round predictions are computed for the 10 % left out. To compute the metrics, those predictions are pooled.

# Implementation of k-fold cross-validation

- Recommended $k = 10$ or $k = 5$ for time-consuming model fits
- $k = n$ is possible (leave-one-out cross-validation), but not recommended due to spatial auto-correlation
- Splitting by spatial clusters/blocks possible, e.g. to evaluate spatial extrapolation capacity (`cv(.., method = "block")` in R package `georob` offers this option). But, not recommended in general because metrics become too pessimistic.
- To allow for comparison of different models cross-validation sets have to be kept fixed.
- Repeated cross-validation with repeated splitting of data: Allows to obtain variation of metrics.
- Combine with data-splitting: 20 % for validation, 80 % for model selection with cross-validation.

# k-fold cross-validation – Problems

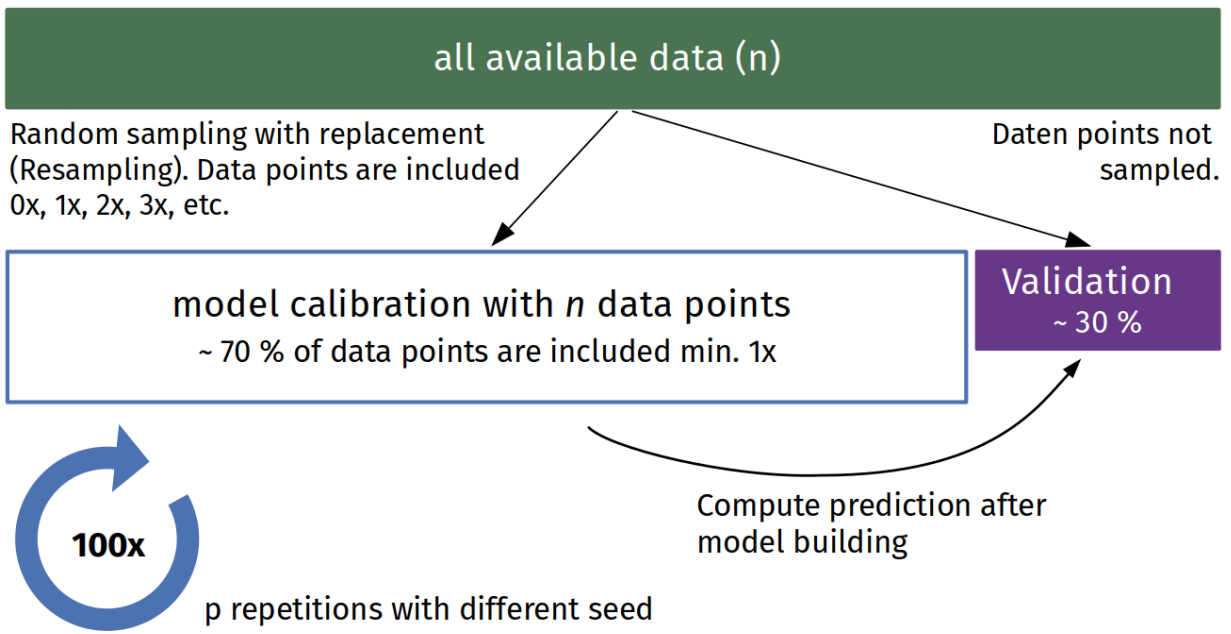
- Random splitting: same problem as with data splitting. If available data does not well represent target area/population, validation results might be biased.
- If cross-validation has been used for model selection, the final cross-validation results do not yield independent model performance metrics anymore.

Solution: nested cross-validation, inner loop for model selection, outer loop for model evaluation.

Example of wrong use of cross-validation:

- Pre-selection of covariates based on correlation coefficient (using all data)
- Create $k$ subsets, fit models by leaving each $k$ subsets out in turn
- Compute predictions for left out subsets, compute error metrics

# Non-parametric bootstrap




Problems:

- Random splitting of biased data: same problem as for data-splitting or cross-validation.
- Larger number of models needed.

Potential: simulate standard errors or prediction intervals for models that do not provide these.

Bootstrap corresponds to **out-of-bag** predictions in random forest.

# 3.3 Model assessment metrics and displays

## Interval scaled response – validation scatterplot

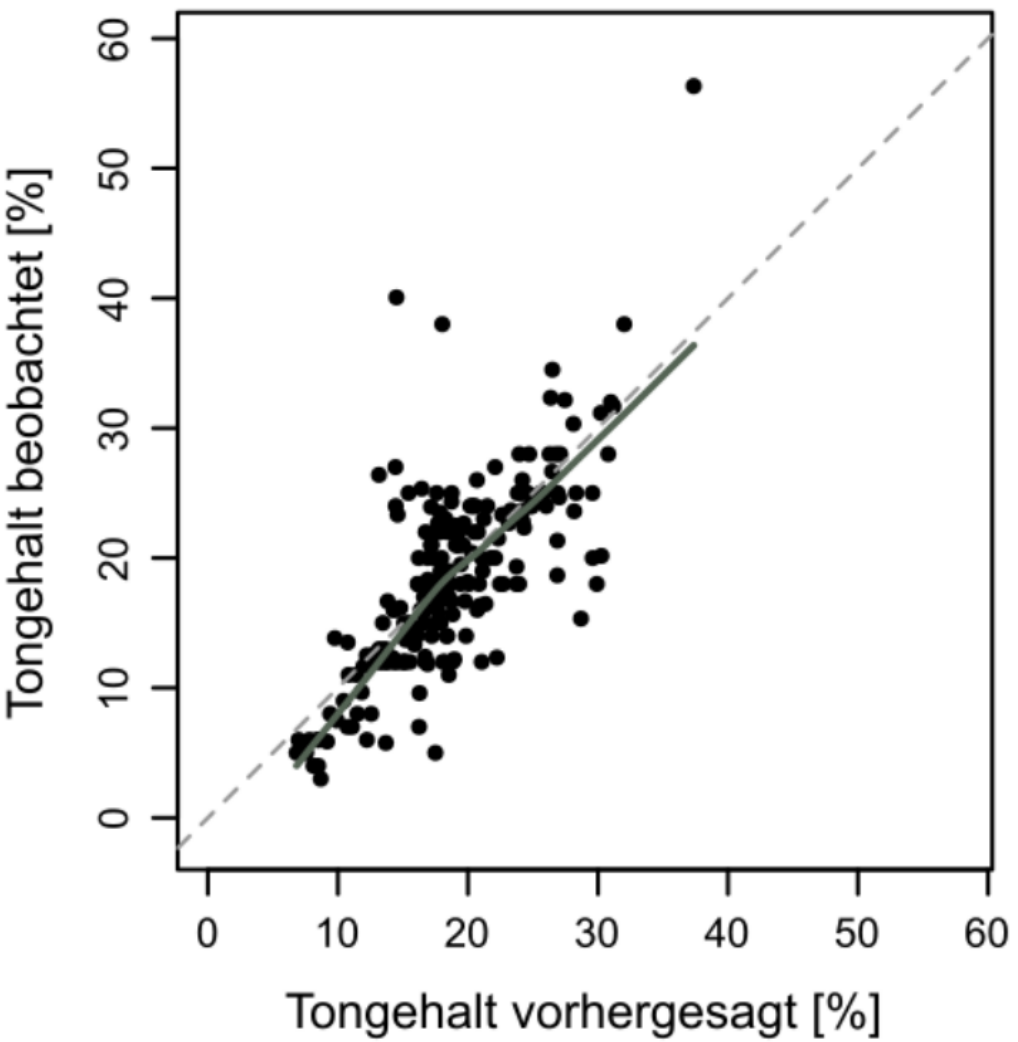
**y axis**: observed
**x axis**: predicted

**1:1 line** indicating perfect prediction

**Lowess scatterplot smoother**
indicates conditional bias

**Regression line**: rather not

**Dots in scatterplot**
→ check distribution, what went wrong?
→ check outliers



# Interval scaled response – metrics

To summarized the validation scatterplot, it is recommended to report at least 3 metrics:

- a bias metric, indicating marginal bias
- an overall precision statistic, reporting the standard deviation of the errors in the original unit of the response (bias and random variation)
- a standardized error statistic, comparable across datasets.

Often reported are mean error (ME) or bias, root mean squared error (RMSE) and $R^2$.

# Interval scaled response – metrics Bias and RMSE

With $y_i$ being the observed value at location $i$ and $\hat{Y}(\mathbf{x}_i)$ the predicted map at location $i$.

Bias (positive: over-prediction, negative: under-prediction):

$$\mathrm{BIAS} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}(\mathbf{x}_i) - y_i\right)$$

Root mean square error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}(\mathbf{x}_i) - y_i\right)^2}$$

# Interval scaled response – metrics $R^2$

$R^2$ (also model efficiency coefficient or mean squared error skill score) with $\overline{y}$ being the mean of the observed data

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n} \left( y_i - \hat{Y}(\mathbf{x}_i) \right)^2}{\sum_{i=1}^{n} \left( y_i - \overline{y} \right)^2}$$

1: perfect prediction.

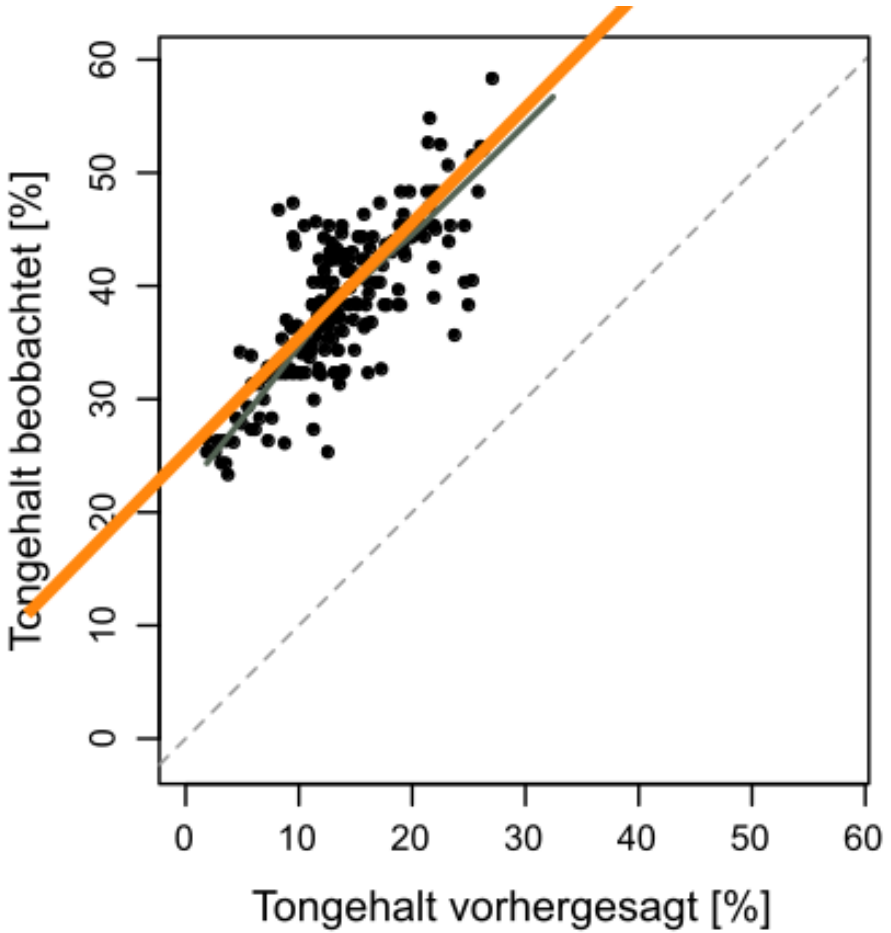
0: mean squared error is the same as variance of observed data, i.e. a map showing the mean of the data would have the same average prediction performance.

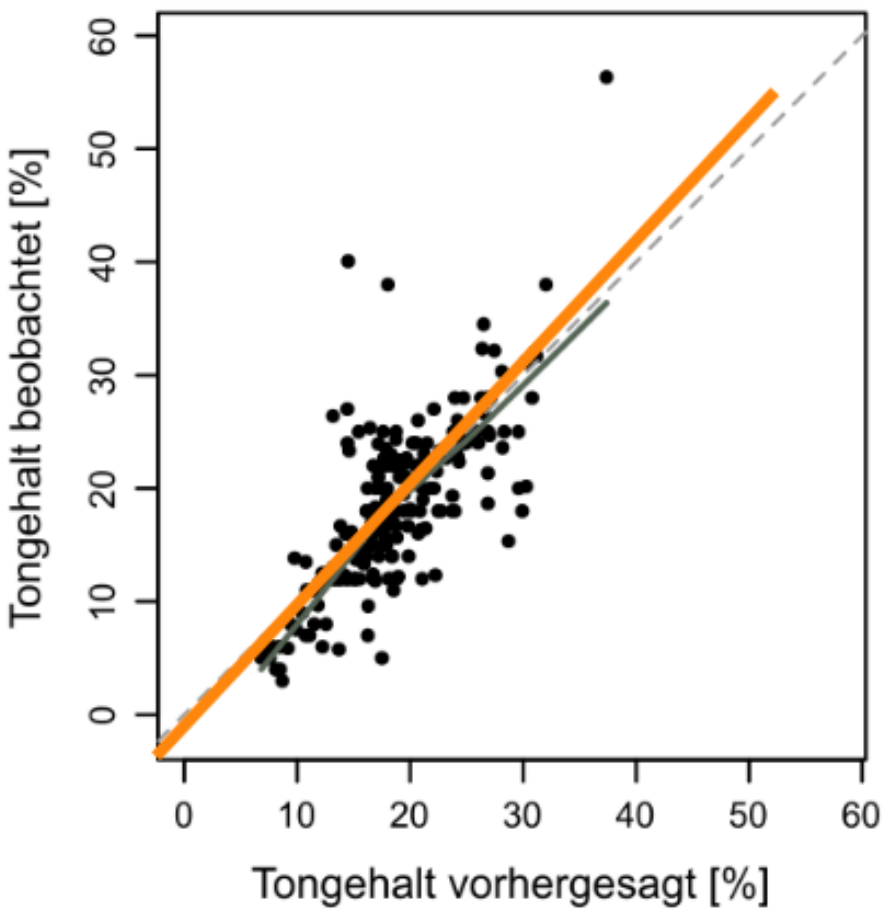
<0: mean squared error is larger than variance of observed data.

# Interval scaled response – $R^2$ vs. $r^2$

$R^2 \neq r^2$.

$R^2$ does not correspond to Pearson correlation coefficient in presence of bias.



# Binary responses – confusion matrix

## Some metrics

(Wilks 2011, Statistical Methods in the Atmospheric Sciences, p. 308 ff)

Percentage correct = (a+d) / n

Hit rate = a / (a+c)

False alarm rate = b / (a+b)

Bias ratio = (a+b) / (a+c)

R-Package «verification»
`multi.cont()`

| | | Tornados Observed | |
|---|---|---|---|
| | | Yes | No |
| Tornados | Yes | **28** (a) | **72** (b) |
| Forecast | No | **23** (c) | **2680** (d) |

n=2803

# Binary responses – confusion matrix

| **1** | | Tornados Observed | | **2** | | Tornados Observed | |
|---|---|---|---|---|---|---|---|
| | | Yes | No | | | Yes | No |
| Tornados | Yes | **28** | **72** | Tornados | Yes | **0** | **0** |
| Forecast | No | **23** | **2680** | Forecast | No | **51** | **2752** |
| | | n=2803 | | | | n=2803 | |

| | **Model 1** | **Model 2** |
|---|---|---|
| **Percentage correct** | 0.97 | 0.98 |
| **False alarm rate** | 0.72 | 0.0 |
| **HSS** | 0.355 | 0 |
| **PSS** | 0.523 | 0 |

Hedging / “gaming” the score: Uninformative predictions yield good scores. Alternative: Skill scores for confusion matrices, e.g.:

- HSS: Heidke skill score (also kappa)
- PSS: Pierce skill score

# Binary responses – Pierce skill score

Pierce skill score (simplified notation):

$$PSS = \frac{ad - bc}{(a + c)(b + d)}$$

Reference: random predictions that are constrained to be unbiased.

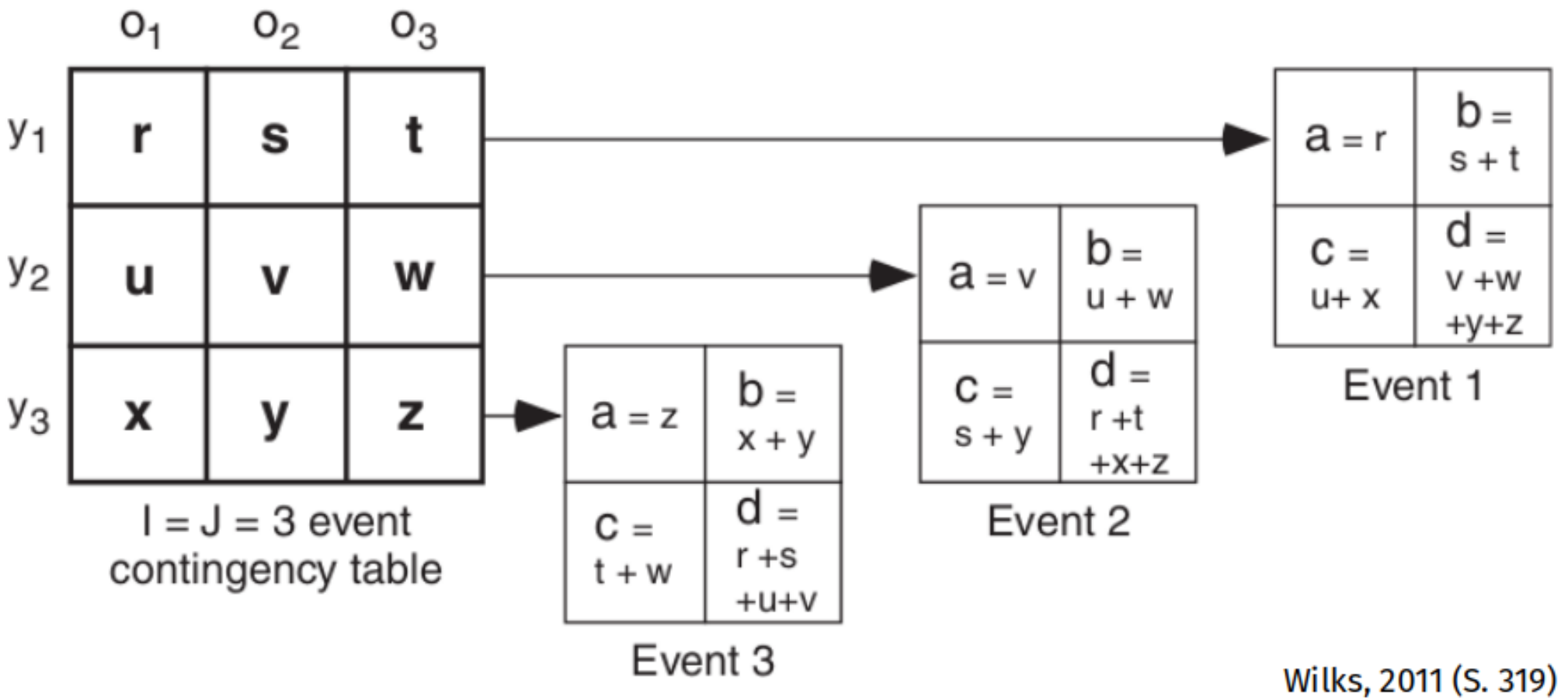
Score range $[-1, 1]$:

1: perfect prediction

0: prediction equal to random class assignment

-1: prediction always opposite of observed class

# Multinomial responses – extension of binary case




Wilks, 2011 (S. 319)

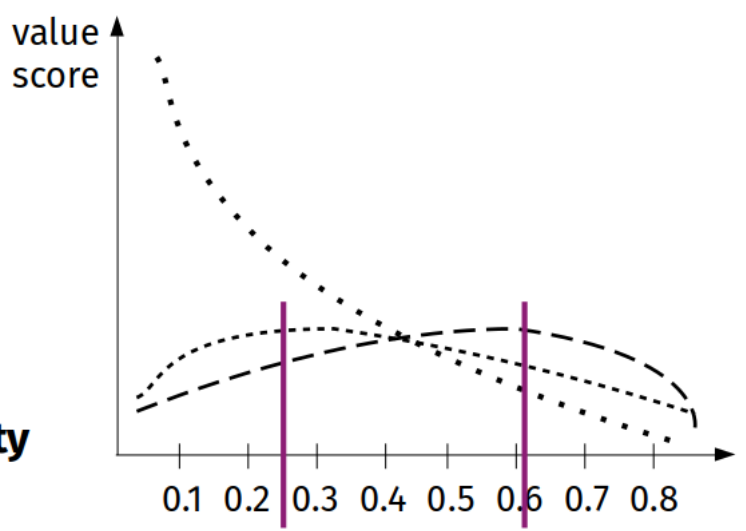
# Use score to optimally discretize predictions

Rainfall occurence at specific time point [1 = yes, 0 = no]
(binary, with predicted probabilities)

| observed | predicted probability | predicted class |
|---|---|---|
| 1 | 0.8 | 1 |
| 0 | 0.7 | 1 |
| 1 | 0.9 | 1 |
| 0 | 0.1 | 0 |
| ... | ... | |

**Typically** – optimize for "percentage correct"
>0.5 → class 1
<0.5 → class 0

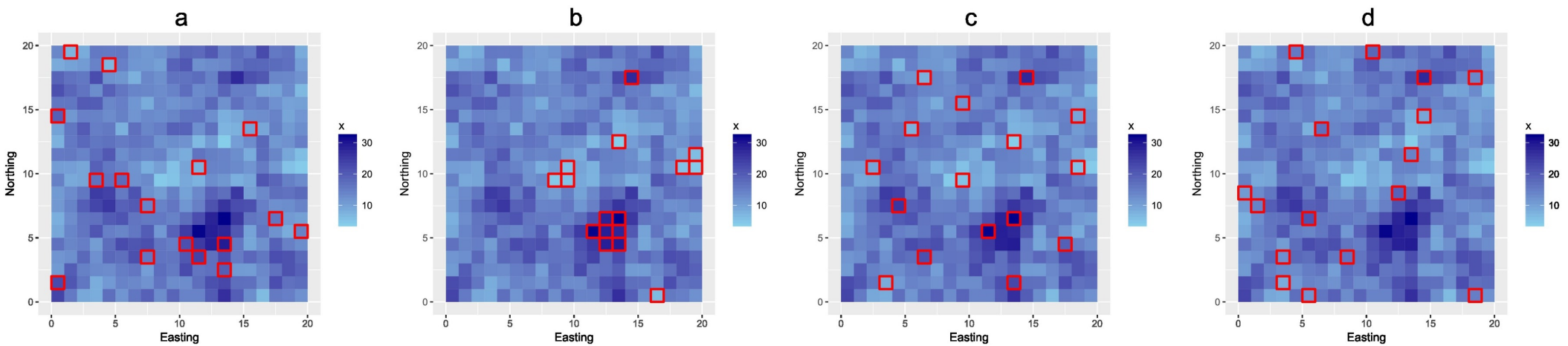
**Better: Optimize on a score with desired property**



# 4 Outlook on more spatial statistics topics

# 4.1 Spatial sampling desings

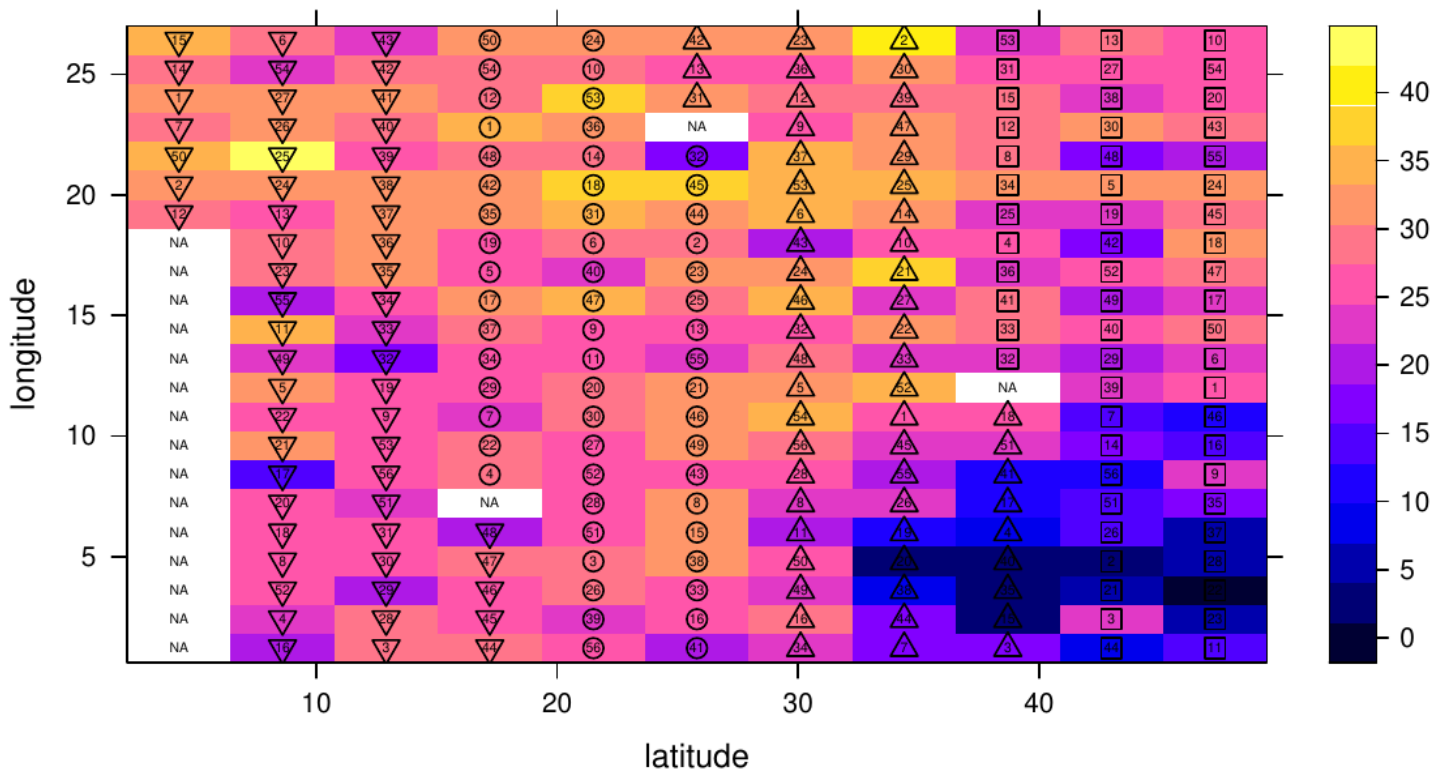
- Model-based sampling for calibration: Optimal selection of sampling location depends on modelling approach used.
- Design-based sampling for validation: random sampling design.




From left to right: simple random sample (a), optimized sample for mapping with linear regression model (b), optimized sample for kriging with an external drift (c), and stratified sample using sixteen equal-sized covariate strata (d). All samples are plotted in a map of the covariate (Brus, 2018).

# 4.2 Geostatistical analysis of experimental data

Example: Field experiment to compare the yield of 56 varieties of wheat in a randomized block experiment.




color scale: clay content

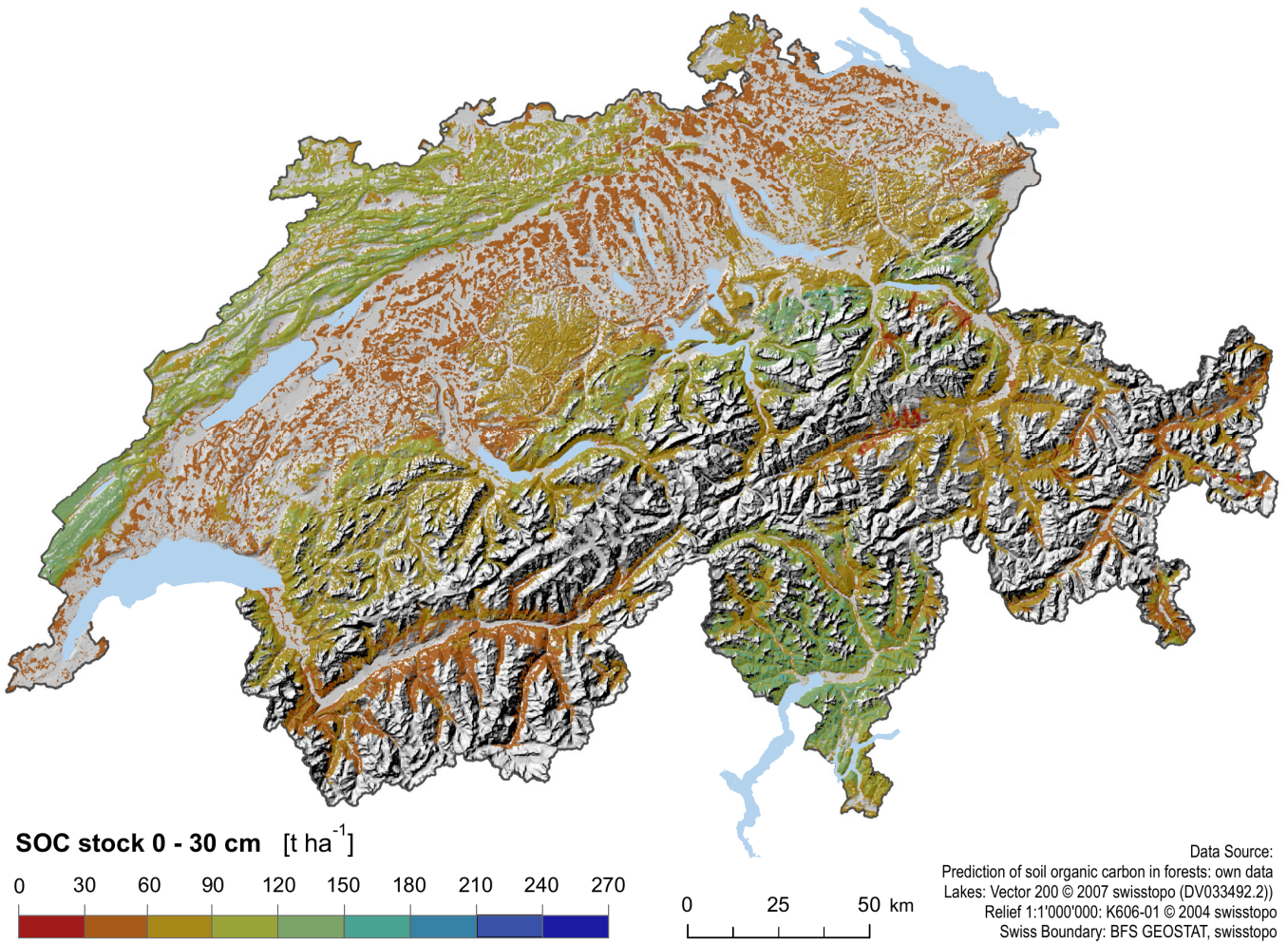
# Geostatistical analysis of experimental data

- Field experiments in ecology, agriculture, forestry, often give rise to spatial data
- Classical analysis of variance of spatial experimental data ignores spatial structure of the data
- Blocking and randomisation sometimes not effective to account for natural heterogeneity within experimental site
- Residuals often violate independence assumption of classical analysis of variance methods
- Explicit consideration of auto-correlation by generalized least squares estimation.

# 4.3 Block kriging

Mean and standard error estimates for finite support targets, e.g. parcels, municipalities.

**Example**: Soil organic carbon stocks, mapped with robust external-drift kriging for Swiss forests.

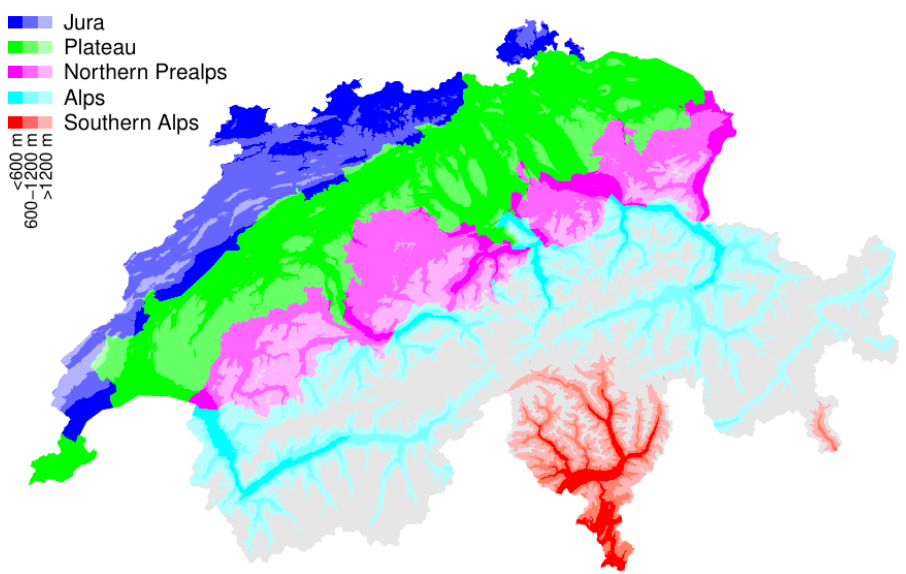


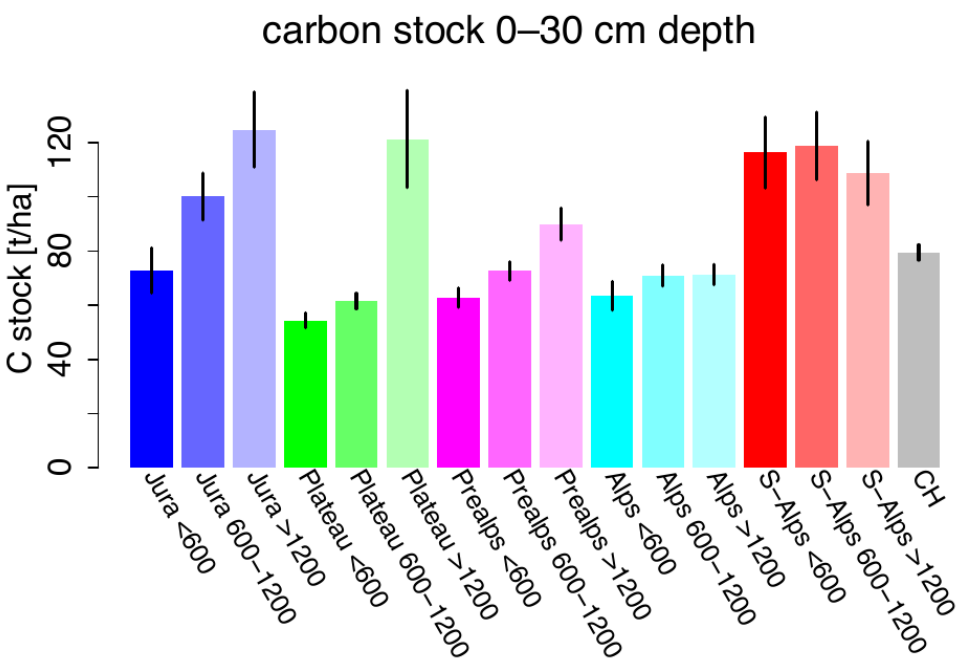
# Block kriging SOC stocks

Basis: Punctual kriging and fitted (residual) variogram.

**Mean per unit**: aproximated by mean of punctual kriging predictions.

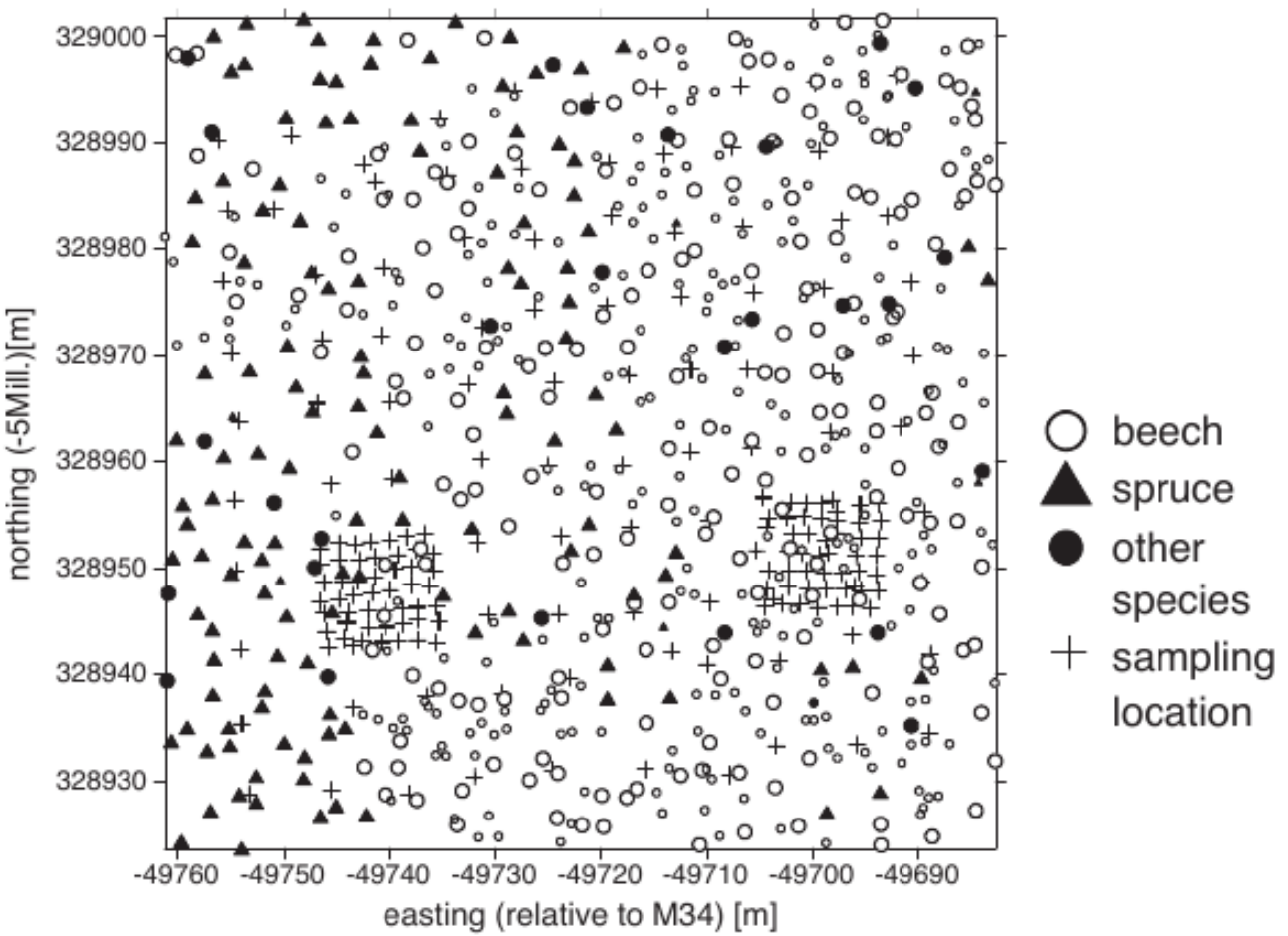
**Standard errors**: “correct” for spatial covariance structure modeled by variogram. I.e. the larger the spatial auto-correlation, the smaller the standard errors for a finite target.



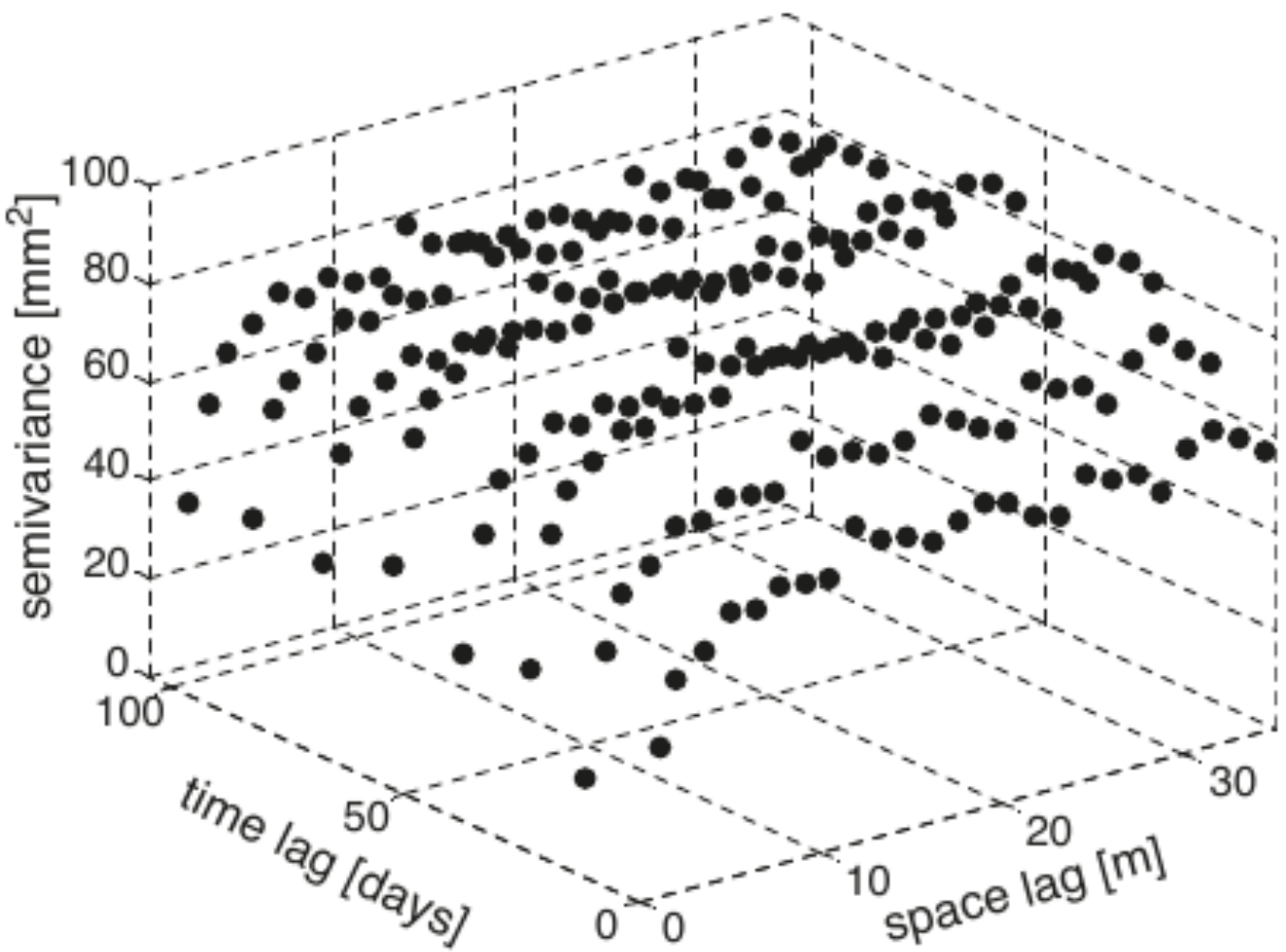



# 4.4 Space-time data

Temporal change of soil water storage in a forest. Measurement of soil water storage biweekly during growing season for 2 years (Jost et al., 2005)



# Space-time sample variogram



# Space-time sample variogram

Product-sum covariance model fitted to space-time sample variogram.