ISRIC Spring School 2017

Regression and machine learning methods for DSM

ISRIC World Soil Information

Bas Kempen

Content

- Part 1: Regression kriging with a linear trend model:
  - linear regression
  - model assumptions
  - model selection
  - uncertainty assessment
  - transformation and back-transformation

- Part 2: Regression kriging with random forests
  - growing a forest of trees
  - out-of-bag accuracy assessment
  - variable importance

- R examples

Part 1: Regression kriging with a linear trend model

ISRIC World Soil Information

Regression kriging

$$Z(\mathbf{s}) = m(\mathbf{s}) + \varepsilon(\mathbf{s})$$

dependent, target variable

trend, explanatory part

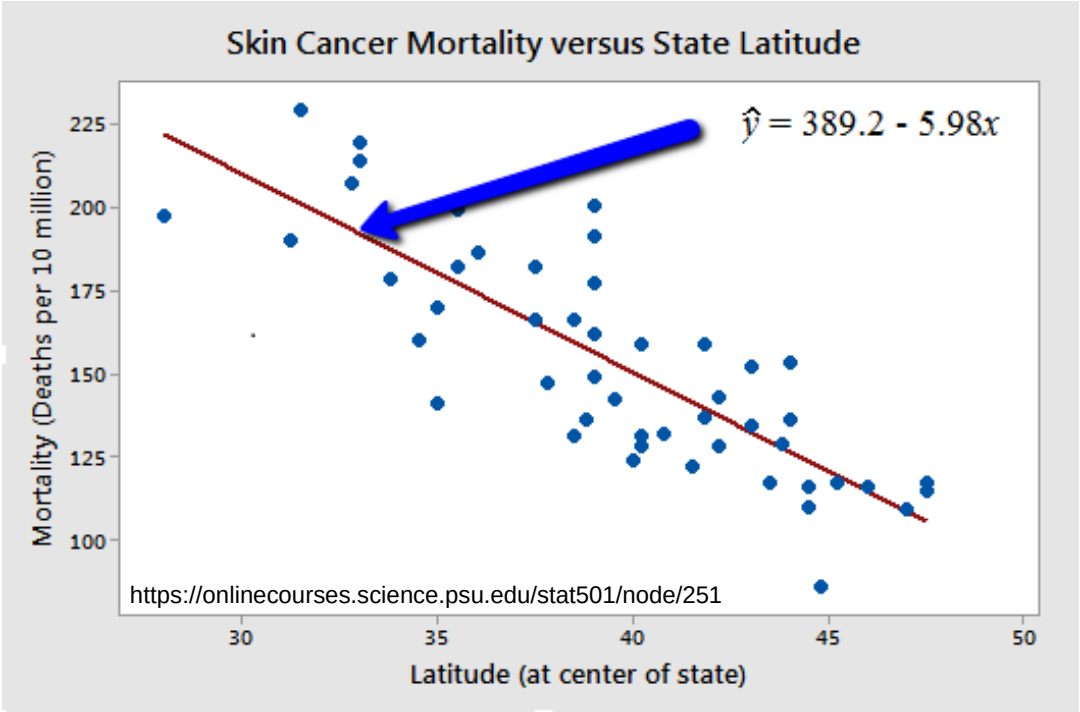
stochastic residual, unexplanatory part, can be spatially correlated!

Unlike ordinary kriging, in regression kriging the trend is no longer constant but a function of 'explanatory' variables, for example:

$$soil\ carbon(\mathbf{s}) = \beta_0 + \beta_1 \cdot elevation(\mathbf{s}) + \beta_2 \cdot slope\ (\mathbf{s}) \ + \beta_3 \cdot NDVI(\mathbf{s}) \ + residual(\mathbf{s})$$

Linear regression

- Statistical method for modelling the relationship between a response variable and one or more explanatory (predictor) variables



Multiple linear regression

- Advantages:
  - Easy interpretation
  - Assessment of prediction uncertainty is straightforward
  - Easy to implement
  - Computationally light

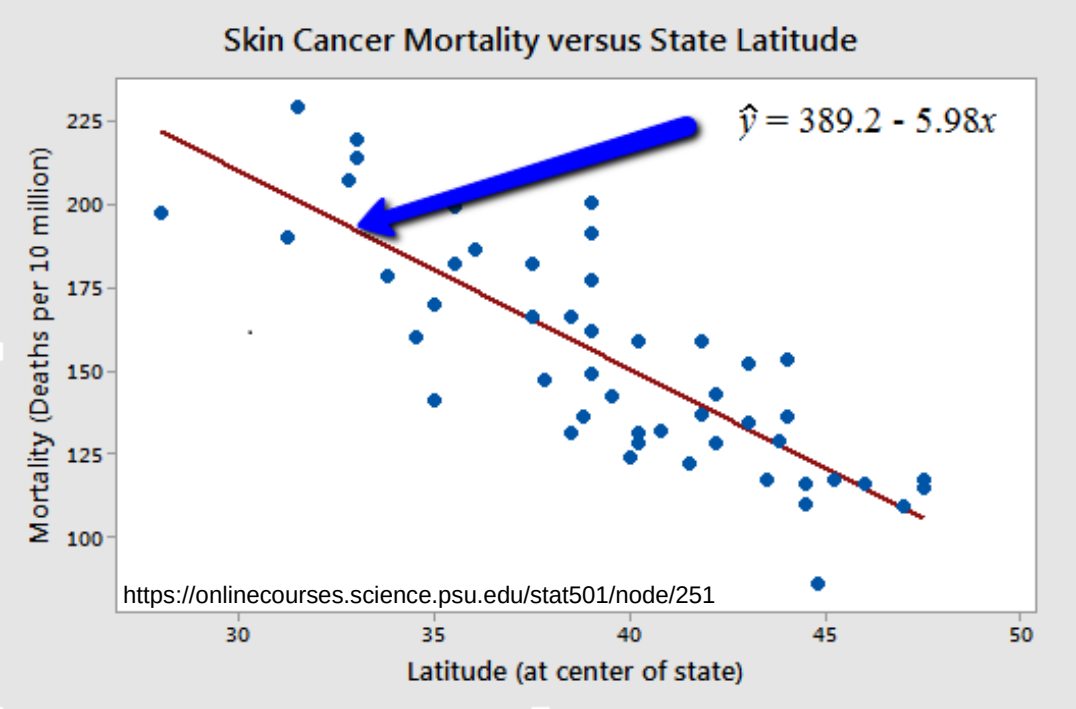
- Parameter estimation with least squares, gives the best linear unbiased estimation.

Multiple linear regression

- Assumptions:
  - Linear relationship (positive/negative) between soil and environmental covariates, additive effects
  - Residuals
    - Independent
    - Constant variance (homoscedacity): often residuals are heteroscedastic: variance increases with fitted value
    - Normal distribution (transformation)
  - Covariates (predictors) are deterministic (assumed to be error free) and uncorrelated

Multiple linear regression

- Assumptions:
  - Linear relationship (positive/negative) between soil and enviro
  - Residu
    - Ind
    - Cor uals are het value -> trar
    - Nor mation
  - Covari umed to be error free) and uncorrelated



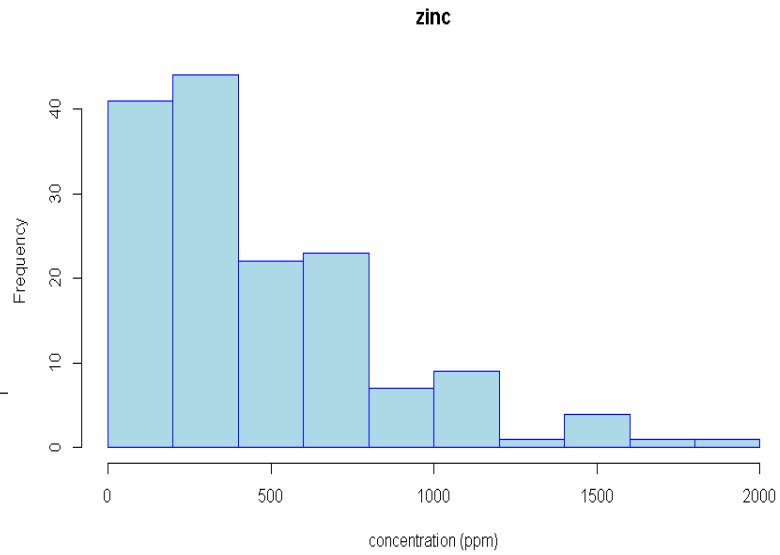
Multiple linear regression

- Assumptions:
  - Linear relationship (positive/negative) between soil and environmental covariates, additive effects
  - Residuals
    - Independent
    - Constant variance (homoscedacity): often residuals are heteroscedastic: variance increases with fitted value -> transformation
    - Normal distribution; non-normality -> transformation
  - Covariates (predictors) are deterministic (assumed to be error free) and uncorrelated

Fitting a linear regression model in R

- lm function (base package)
- R example (meuse data; sp package)

- Zinc
- Covariates: distance to river, elevation, organic matter, soil class, flooding frequency class, lime class
- For categorical covariates, at least one observation is required for each category

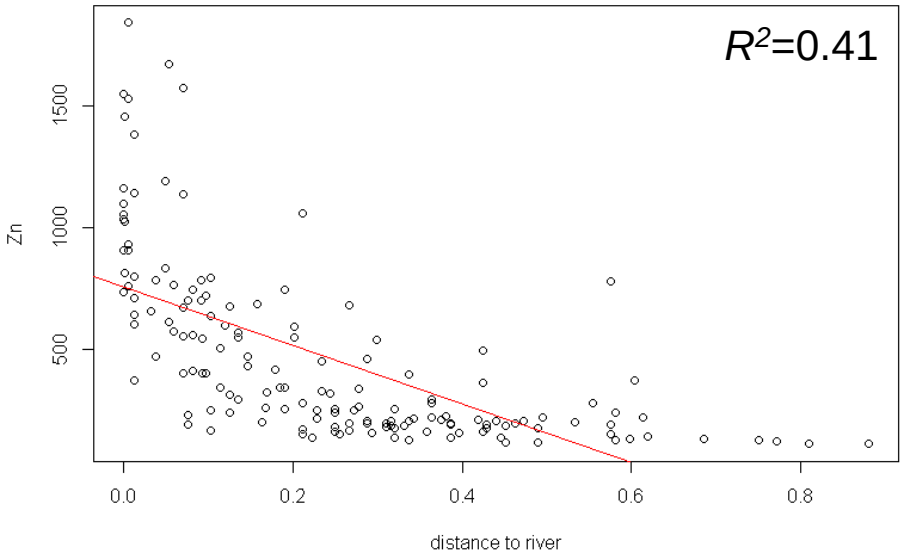


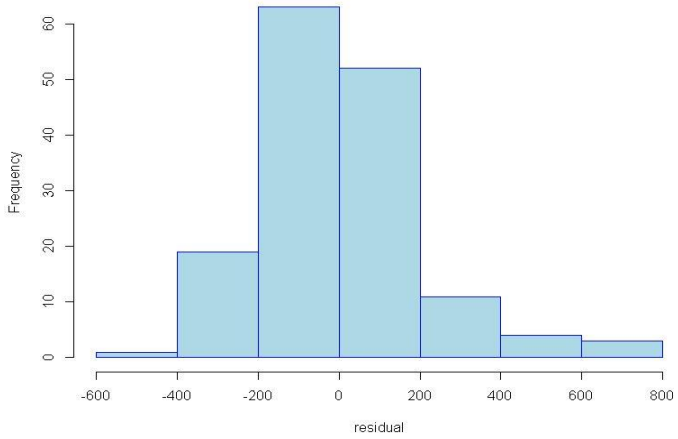
Fitting a linear regression model in R

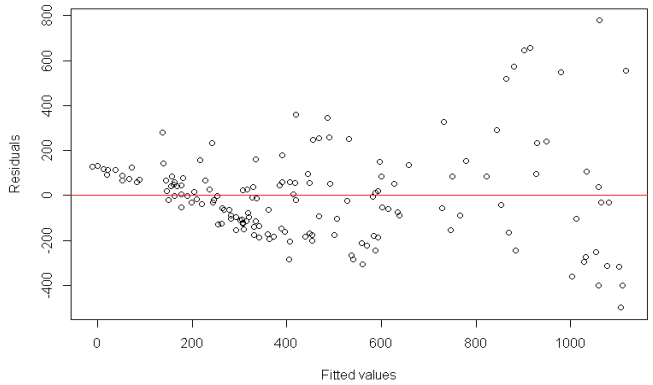
```
# fit a linear regression model
meuse.lm <- lm(zinc ~ dist + elev + om + soil + ffreq + lime + dist.m, data = meuse)
summary(meuse.lm)
```

```
Call:
lm(formula = zinc ~ dist + elev + om + soil + ffreq + lime +
    dist.m, data = meuse)

Residuals:
    Min      1Q  Median      3Q     Max
-499.37 -127.23  -14.51   84.03  777.80

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  982.6359   181.1140   5.426 2.41e-07 ***
dist         375.4192   536.2389   0.700 0.485004
elev         -80.5773    23.1019  -3.488 0.000647 ***
om            36.0134     7.2344   4.978 1.82e-06 ***
soil2          1.5719    57.1215   0.028 0.978085
soil3         59.8260    85.1853   0.702 0.483631
ffreq2      -123.3232    53.1183  -2.322 0.021663 *
ffreq3       -83.7905    63.7460  -1.314 0.190802
lime1        125.7849    55.1698   2.280 0.024089 *
dist.m        -0.6917     0.4593  -1.506 0.134237
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 213.6 on 143 degrees of freedom
Multiple R-squared:  0.6827, Adjusted R-squared:  0.6627
F-statistic: 34.18 on 9 and 143 DF,  p-value: < 2.2e-16
```

Checking model assumptions







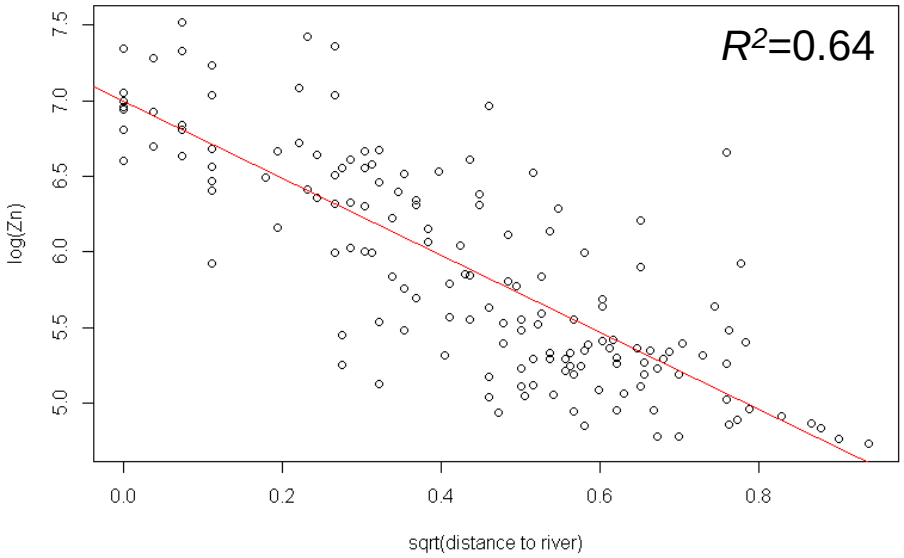
- Transform: log, sqrt
- Convert to categorical: quantile splitting

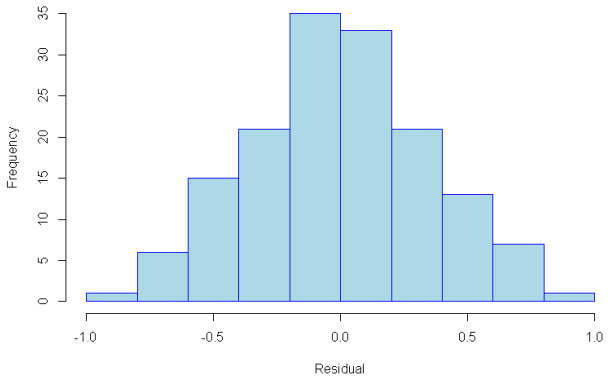
Log-transform the Zinc content

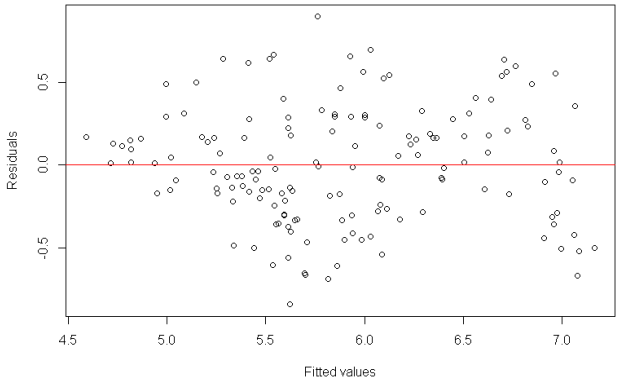
```
# fit a linear regression model
meuse.lm <- lm(log(zinc) ~ dist + elev + om + soil + ffreq + lime + dist.m, data = meuse)
```

```
> summary(meuse.lm)

Call:
lm(formula = log(zinc) ~ dist + elev + om + soil + ffreq + lime +
    dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84108 -0.24385 -0.01372  0.23050  0.89395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4052722  0.3042102  24.343  < 2e-16 ***
dist         0.7347752  0.9006996   0.816 0.415981
elev        -0.1782247  0.0388034  -4.593 9.50e-06 ***
om           0.0574210  0.0121513   4.725 5.43e-06 ***
soil2       -0.0705382  0.0959447  -0.735 0.463424
soil3       -0.0010691  0.1430824  -0.007 0.994048
ffreq2      -0.3093408  0.0892207  -3.467 0.000695 ***
ffreq3      -0.2448201  0.1070716  -2.287 0.023693 *
lime1        0.0355855  0.0926666   0.384 0.701536
dist.m      -0.0017806  0.0007714  -2.308 0.022422 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3589 on 143 degrees of freedom
Multiple R-squared:  0.7673, Adjusted R-squared:  0.7527
F-statistic: 52.39 on 9 and 143 DF,  p-value: < 2.2e-16
```

Checking model assumptions







Log-transform the Zinc content

```
# fit a linear regression model
meuse.lm <- lm(log(zinc) ~ dist + elev + om + soil + ffreq + lime + dist.m, data = meuse)
```

```
> summary(meuse.lm)

Call:
lm(formula = log(zinc) ~ dist + elev + om + soil + ffreq + lime +
    dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84108 -0.24385 -0.01372  0.23050  0.89395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4052722  0.3042102  24.343  < 2e-16 ***
dist         0.7347752  0.9006996   0.816 0.415981
elev        -0.1782247  0.0388034  -4.593 9.50e-06 ***
om           0.0574210  0.0121513   4.725 5.43e-06 ***
soil2       -0.0705382  0.0959447  -0.735 0.463424
soil3       -0.0010691  0.1430824  -0.007 0.994048
ffreq2      -0.3093408  0.0892207  -3.467 0.000695 ***
ffreq3      -0.2448201  0.1070716  -2.287 0.023693 *
lime1        0.0355855  0.0926666   0.384 0.701536
dist.m      -0.0017806  0.0007714  -2.308 0.022422 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3589 on 143 degrees of freedom
Multiple R-squared:  0.7673, Adjusted R-squared:  0.7527
F-statistic: 52.39 on 9 and 143 DF,  p-value: < 2.2e-16
```

Model selection

- Selecting a statistical model from a set of candidate models; not trivial
- Select the best (most parsimonious) model; Occam's razor: "among competing hypotheses that predict equally well, the one with the fewest assumptions should be selected".
- Selection criterion: Akaike Information Criterion (AIC): goodness of fit of the model, penalty for use of variables

$$AIC = -2\ln(L) + 2k$$

- The smaller the (absolute) AIC value the better.

Stepwise selection in R

- Use stepwise selection to fit a more parsimonious model

```
# stepwise selection of covariates
meuse.lm2 <- step(lm(log(zinc) ~ dist + elev + om + soil + ffreq + lime + dist.m, data = meuse))
```

```
> summary(meuse.lm2)

Call:
lm(formula = log(zinc) ~ elev + om + ffreq + dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.76235 -0.26803 -0.01208  0.24639  0.88375

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4020309  0.2855998  25.917  < 2e-16 ***
elev        -0.1865056  0.0352420  -5.292 4.31e-07 ***
om           0.0640476  0.0105028   6.098 9.02e-09 ***
ffreq2      -0.2704367  0.0752778  -3.593 0.000446 ***
ffreq3      -0.2199716  0.0896215  -2.454 0.015277 *
dist.m      -0.0011897  0.0001708  -6.965 1.02e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3561 on 147 degrees of freedom
Multiple R-squared:  0.7644, Adjusted R-squared:  0.7564
F-statistic: 95.38 on 5 and 147 DF,  p-value: < 2.2e-16
```

```
> summary(meuse.lm)

Call:
lm(formula = log(zinc) ~ dist + elev + om + soil + ffreq + lime +
    dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84108 -0.24385 -0.01372  0.23050  0.89395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4052722  0.3042102  24.343  < 2e-16 ***
dist         0.7347752  0.9006996   0.816 0.415981
elev        -0.1782247  0.0388034  -4.593 9.50e-06 ***
om           0.0574210  0.0121513   4.725 5.43e-06 ***
soil2       -0.0705382  0.0959447  -0.735 0.463424
soil3       -0.0010691  0.1430824  -0.007 0.994048
ffreq2      -0.3093408  0.0892207  -3.467 0.000695 ***
ffreq3      -0.2448201  0.1070716  -2.287 0.023693 *
lime1        0.0355855  0.0926666   0.384 0.701536
dist.m      -0.0017806  0.0007714  -2.308 0.022422 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3589 on 143 degrees of freedom
Multiple R-squared:  0.7673, Adjusted R-squared:  0.7527
F-statistic: 52.39 on 9 and 143 DF,  p-value: < 2.2e-16
```

Model selection: AIC

- Compare models with AIC

```
# AIC
AIC(meuse.lm)
AIC(meuse.lm2)
```

```
> AIC(meuse.lm)
[1] 135.8478
> AIC(meuse.lm2)
[1] 131.5968
```

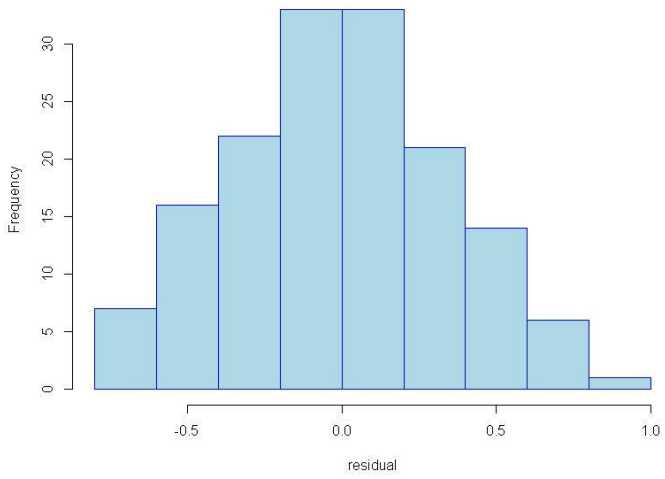
```
> summary(meuse.lm)

Call:
lm(formula = log(zinc) ~ dist + elev + om + soil + ffreq + lime +
    dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84108 -0.24385 -0.01372  0.23050  0.89395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4052722  0.3042102  24.343  < 2e-16 ***
dist         0.7347752  0.9006996   0.816 0.415981
elev        -0.1782247  0.0388034  -4.593 9.50e-06 ***
om           0.0574210  0.0121513   4.725 5.43e-06 ***
soil2       -0.0705382  0.0959447  -0.735 0.463424
soil3       -0.0010691  0.1430824  -0.007 0.994048
ffreq2      -0.3093408  0.0892207  -3.467 0.000695 ***
ffreq3      -0.2448201  0.1070716  -2.287 0.023693 *
lime1        0.0355855  0.0926666   0.384 0.701536
dist.m      -0.0017806  0.0007714  -2.308 0.022422 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3589 on 143 degrees of freedom
Multiple R-squared:  0.7673, Adjusted R-squared:  0.7527
F-statistic: 52.39 on 9 and 143 DF,  p-value: < 2.2e-16
```

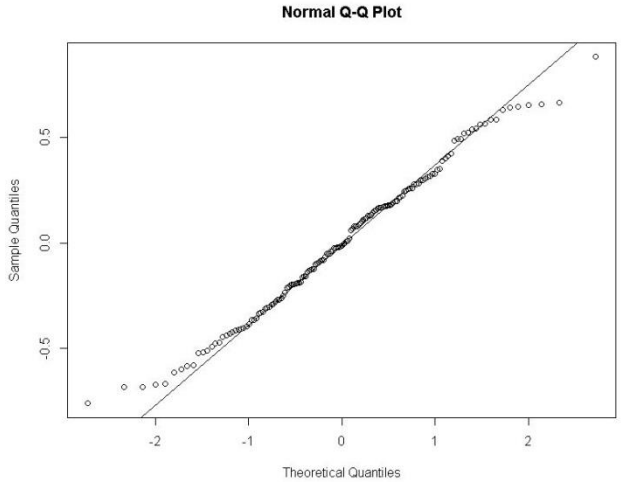
```
> summary(meuse.lm2)

Call:
lm(formula = log(zinc) ~ elev + om + ffreq + dist.m, data = meuse)

Residuals:
     Min       1Q   Median       3Q      Max
-0.76235 -0.26803 -0.01208  0.24639  0.88375

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.4020309  0.2855998  25.917  < 2e-16 ***
elev        -0.1865056  0.0352420  -5.292 4.31e-07 ***
om           0.0640476  0.0105028   6.098 9.02e-09 ***
ffreq2      -0.2704367  0.0752778  -3.593 0.000446 ***
ffreq3      -0.2199716  0.0896215  -2.454 0.015277 *
dist.m      -0.0011897  0.0001708  -6.965 1.02e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3561 on 147 degrees of freedom
Multiple R-squared:  0.7644, Adjusted R-squared:  0.7564
F-statistic: 95.38 on 5 and 147 DF,  p-value: < 2.2e-16
```

Model diagnostics – normal distribution

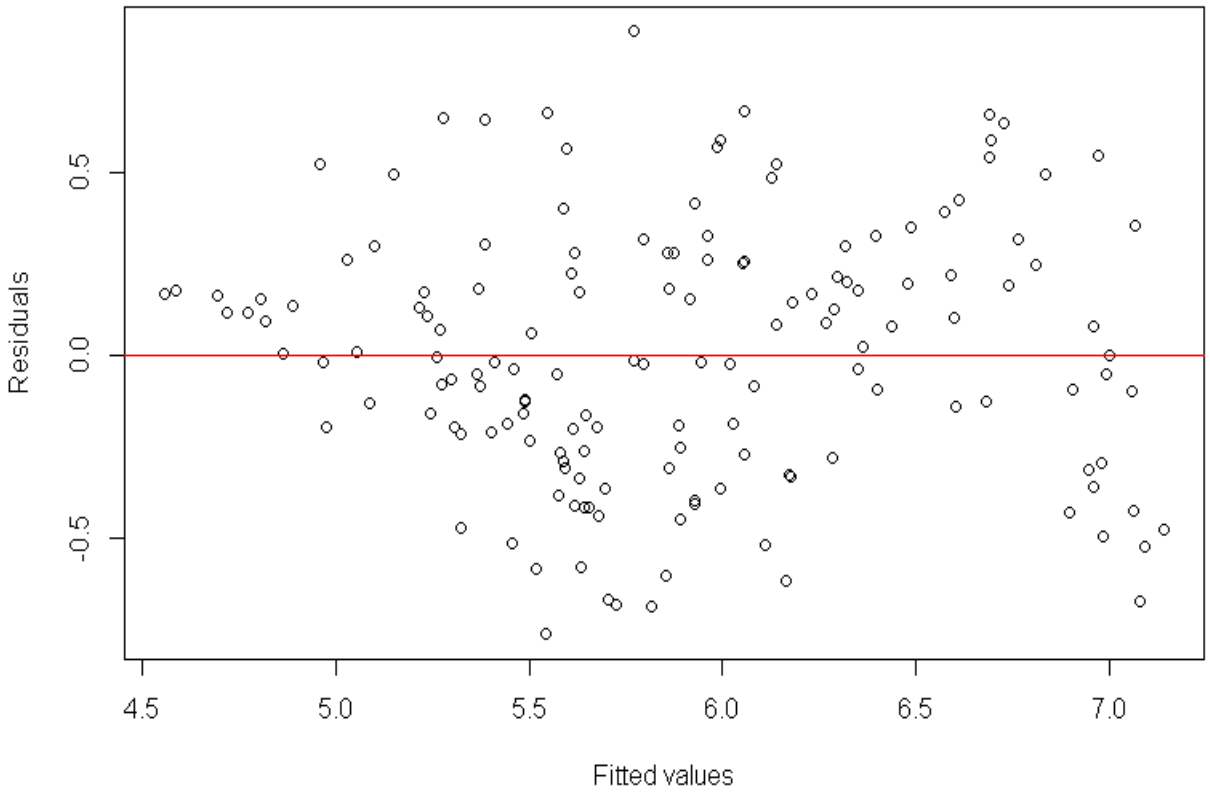
```
# histogram
hist(resid(meuse.lm2),col = "lightblue", border = "blue", main = "residuals",xlab="residual")

# QQplot
qqnorm(resid(meuse.lm2))
qqline(resid(meuse.lm2))

# skewness statistic
skewness(resid(meuse.lm2))
```

```
> skewness(resid(meuse.lm2))
[1] -0.01052761
```

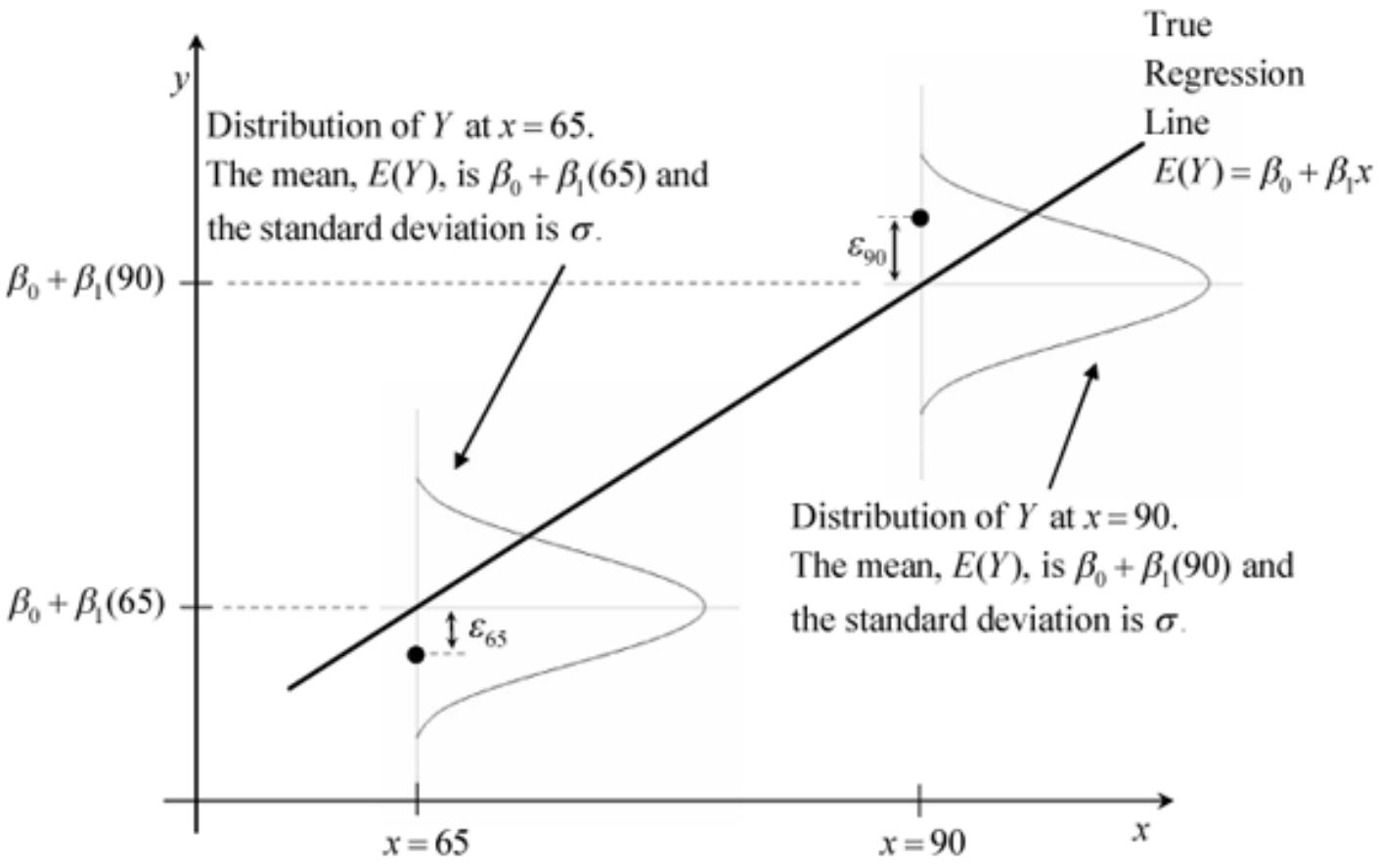




Model diagnostics – constant variance



Linear model: y = E(Y) + ε / ŷ = E(Y)



Reliawiki.org

Prediction uncertainty

- Prediction uncertainty has two components:
  - Uncertainty about the model as result of noise in the data: *residual variance*
  - Uncertainty about the mean: *standard error (or variance) of the mean*

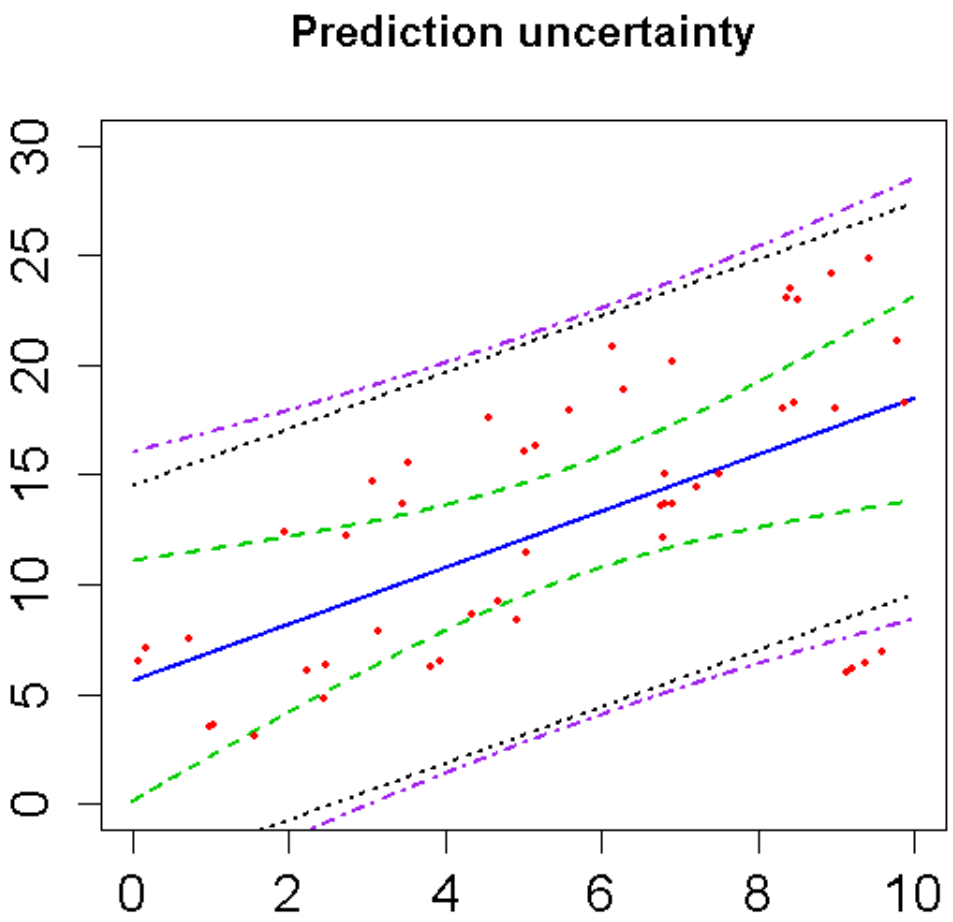
- The prediction uncertainty (prediction error variance, standard error of prediction) is the sum of these components:

$$SE_{pred} = \sqrt{s^2 + SEmean^2}$$

- 90% prediction interval:

$$PI(\mathbf{s}) = \bar{Z}(\mathbf{s}) \pm 1.645 \times SE_{pred}(\mathbf{s})$$

Graphical



SE of prediction

SE of model

SE of mean

Mean: E[Y|X]

Prediction uncertainty assessment in R

```
> # predict with standard errors
> p <- predict.lm(meuse.lm2, data = meuse, se.fit=TRUE)
> str(p)
List of 4
 $ fit           : Named num [1:153] 6.74 6.96 6.6 6.17 6.11 ...
  ..- attr(*, "names")= chr [1:153] "1" "2" "3" "4" ...
 $ se.fit        : num [1:153] 0.0644 0.064 0.0598 0.0399 0.0502 ...
 $ df            : int 147
 $ residual.scale: num 0.356

> # 95% confidence interval of the mean (expected value)
> ci.l <- p$fit - 1.96*p$se.fit; ci.u <- p$fit + 1.96*p$se.fit
> head(round(ci.l,4)); head(round(ci.u,4))
     1      2      3      4      5      6
6.6123 6.8352 6.4842 6.0873 6.0137 5.7823
     1      2      3      4      5      6
6.8647 7.0860 6.7187 6.2437 6.2105 5.9964
```

$SE_{pred} = \sqrt{s^2 + SEmean^2}$

```
> # compute standard error of prediction (prediction standard deviation)
> u <- sqrt(p$se.fit**2+p$residual.scale**2)
> head(round(u,4))
[1] 0.3619 0.3618 0.3611 0.3584 0.3597 0.3603

> # 90% prediction interval
> pi.l <- p$fit - 1.645*u; pi.u <- p$fit + 1.645*u
> head(round(pi.l,4)); head(round(pi.u,4))
     1      2      3      4      5      6
6.1432 6.3654 6.0074 5.5760 5.5205 5.2967
     1      2      3      4      5      6
7.3339 7.5559 7.1955 6.7550 6.7037 6.4821
```

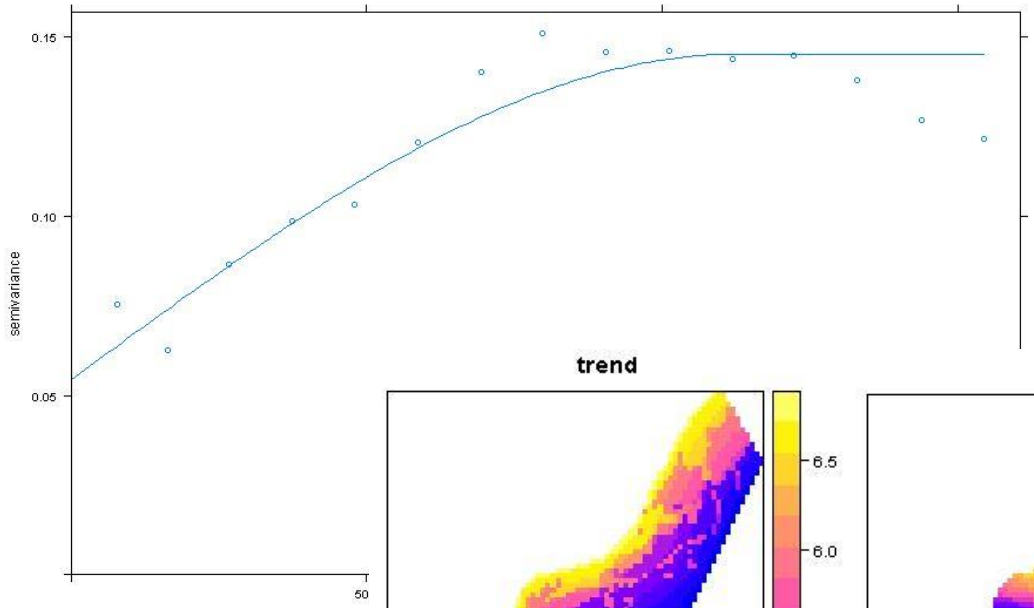
Regression-kriging algorithm

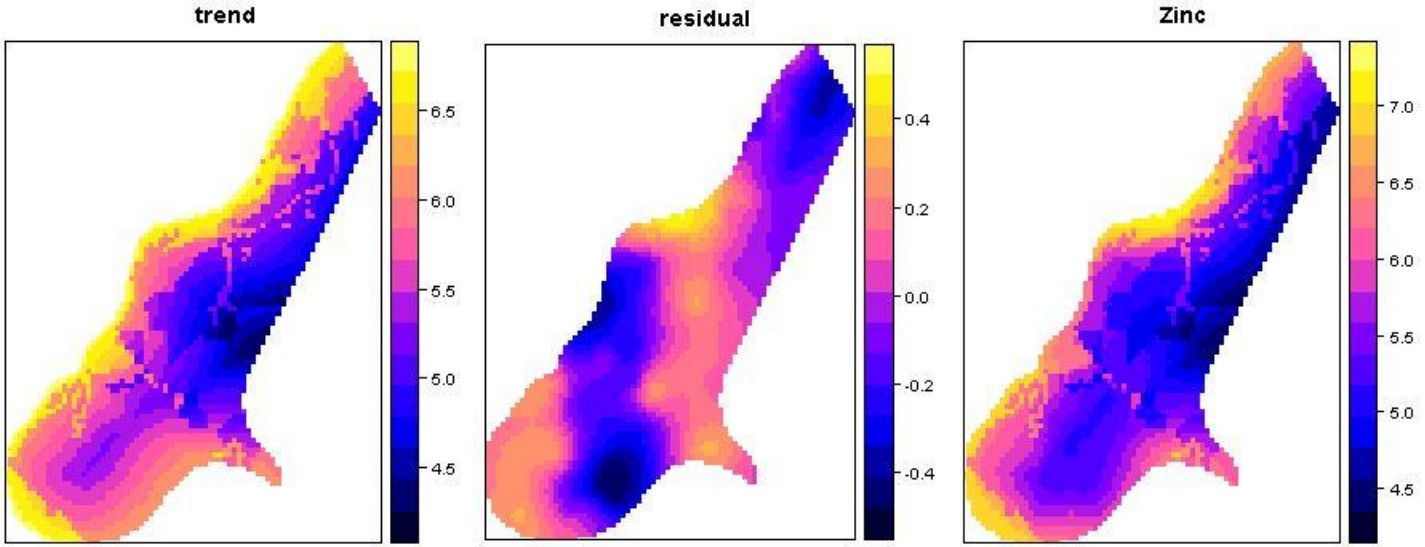
1. select explanatory variables and fit regression model (estimate regression coefficients)
2. compute residuals (by subtracting the fitted trend from the observations) at observation locations and compute from these a semivariogram
3. apply the regression model to all unobserved locations (usually a grid)
4. krige the residuals
5. add up the results of steps 3 and 4

Regression-kriging in R

```
### regression-kriging ###
library(gstat)
library(sp)

# fit linear model
meuse.fit <- lm(formula = log(zinc) ~ dist + soil + ffreq, data = meuse)

# compute model residual; add to data.frame with point data
meuse$resid <- resid(meuse.fit)

# create a spatial object
coordinates(meuse) <- ~x+y

# model variogram
v <- variogram(resid~1, meuse) # v <- variogram(log(zinc) ~ dist + soil + ffreq, meuse)
vm <- vgm(psill = 0.07, model = "Sph", range = 500, nugget = 0.05)
vmf <- fit.variogram(v, model = vm, fit.ranges = TRUE, fit.sill = TRUE)

# krige residuals
zinc_RK <- krige(
  formula = resid~1,
  locations = meuse,
  newdata = meuse.grid,
  model = vmf
)
summary(zinc_RK)

# predict trend
trend_predict <- predict(meuse.fit,meuse.grid)

# add residuals to trend prediction
zinc_RK$trend <- trend_predict
zinc_RK$logzinc <- zinc_RK$trend + zinc_RK$var1.pred
```

Regression-kriging results





Back-transformation

- The kriging prediction of the Zinc example gives us the prediction on the log-scale.
- For a log-transformed variable taking the exponent does not give the mean of the log-normally distribution (it gives the median).
- Back-transformation of a log-transformed variable:

exp(prediction + 0.5*prediction variance)

- Back-transformation of prediction variance not trivial, depends on predicted value. Lark and Lapworth (2012) argue it is therefore not a good (independent) measure of prediction quality.
- Quantify prediction uncertainty by the 90% prediction interval:
  - Lower boundary: exp(prediction – 1.645 * prediction standard deviation)
  - Upper boundary: exp(prediction + 1.645 * prediction standard deviation)

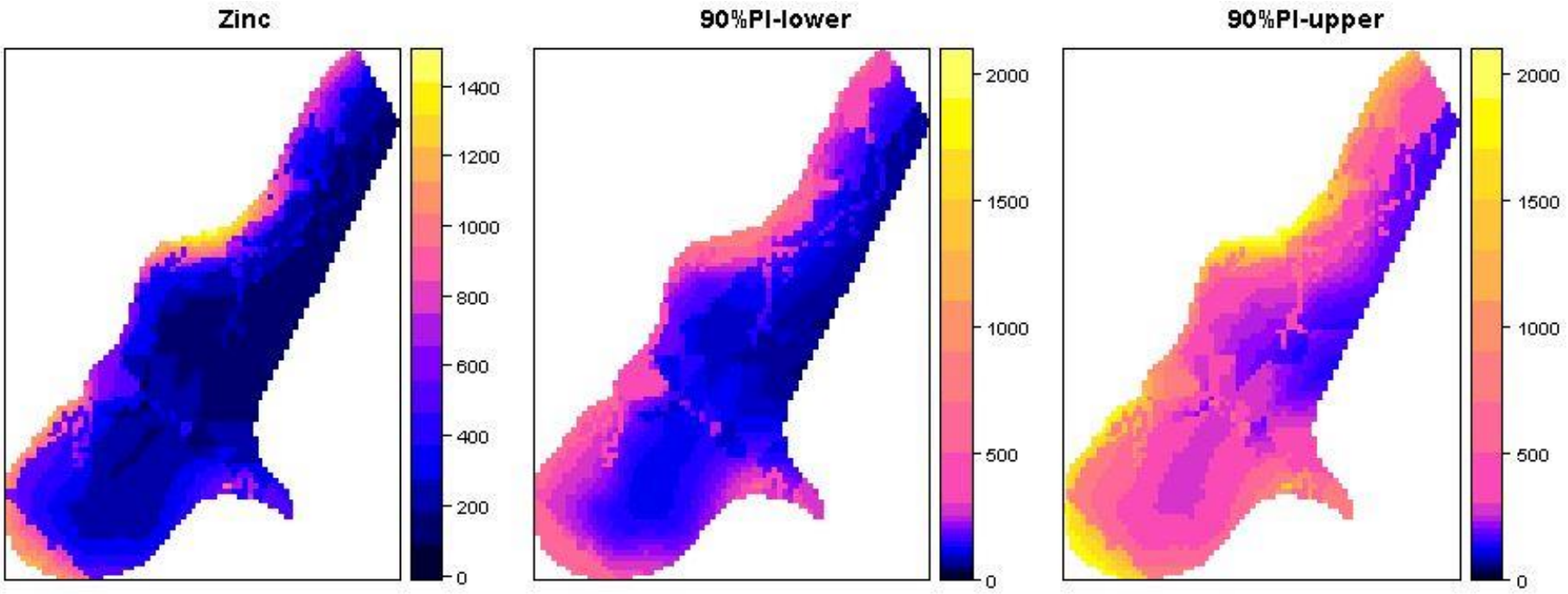
Lark and Lapworth, 2012. Geodema 173-174, 231-240

An example of back-transformation

```
# back-transformation
zinc_predict$zinc <- exp(zinc_predict$logzinc + 0.5 * zinc_predict$var1.var)
zinc_predict$lower <- exp(zinc_predict$logzinc - 1.64 * sqrt(zinc_predict$var1.var))
zinc_predict$upper <- exp(zinc_predict$logzinc + 1.64 * sqrt(zinc_predict$var1.var))
summary(zinc_predict)
```

```
> summary(zinc_predict)
Object of class SpatialPixelsDataFrame
Coordinates:
     min    max
x 178460 181540
y 329620 333740
Is projected: NA
proj4string : [NA]
Number of points: 3103
Grid attributes:
  cellcentre.offset cellsize cells.dim
x            178460       40        78
y            329620       40       104
Data attributes:
   var1.pred           var1.var          trend           logzinc          zinc            lower           upper
 Min.   :-0.483270   Min.   :0.06617   Min.   :4.250   Min.   :4.347   Min.   :  81.7   Min.   : 44.58   Min.   : 132.4
 1st Qu.:-0.168765   1st Qu.:0.07359   1st Qu.:5.092   1st Qu.:5.109   1st Qu.: 172.1   1st Qu.:104.81   1st Qu.: 262.3
 Median :-0.007466   Median :0.07702   Median :5.623   Median :5.504   Median : 256.2   Median :155.19   Median : 391.0
 Mean   :-0.007696   Mean   :0.08041   Mean   :5.604   Mean   :5.596   Mean   : 354.6   Mean   :214.37   Mean   : 541.8
 3rd Qu.: 0.170600   3rd Qu.:0.08446   3rd Qu.:6.015   3rd Qu.:6.049   3rd Qu.: 442.2   3rd Qu.:263.39   3rd Qu.: 675.1
 Max.   : 0.499617   Max.   :0.11909   Max.   :6.718   Max.   :7.218   Max.   :1412.5   Max.   :881.10   Max.   :2123.0
```

An example of back-transformation



Regression-kriging vs. External drift-kriging

- Differ in how the trend is modelled
- Regression-kriging: trend and residuals are modelled separately; uncertainty of the trend not accounted for by the prediction error variance (kriging variance)
- External drift-kriging (universal kriging): trend and residuals are modelled simultaneously
- RK violates the assumption of independent residuals
- KED trend estimates are modelled taking into account spatial correlation
  - Iterative Generalized Least Squares
  - Residual Maximum Likelihood (REML); geoR package

Lark and Cullis, 2004. European Journal of Soil Science 55(4), 799-813
Lark et al., 2006. European Journal of Soil Science 57(6), 787-799
Lark and Webster, 2006. Earth Surface Processes and Landforms 31, 862-874

An R example of KED

```
### external drift-kriging ###
zinc_KED <- krige(
  formula = log(zinc) ~ dist + soil + ffreq,
  locations = meuse,
  newdata = meuse.grid,
  model = vmf
)

# back-transformation
zinc_KED$zinc <- exp(zinc_KED$var1.pred + 0.5 * zinc_KED$var1.var)
zinc_KED$lower <- exp(zinc_KED$var1.pred - 1.64 * sqrt(zinc_KED$var1.var))
zinc_KED$upper <- exp(zinc_KED$var1.pred + 1.64 * sqrt(zinc_KED$var1.var))
```

An R example of KED

```
> summary(zinc_RK)
Object of class SpatialPixelsDataFrame
Coordinates:
     min    max
x 178460 181540
y 329620 333740
Is projected: NA
proj4string : [NA]
Number of points: 3103
Grid attributes:
  cellcentre.offset cellsize cells.dim
x            178460       40        78
y            329620       40       104
Data attributes:
   var1.pred           var1.var          trend           logzinc          zinc            lower           upper
 Min.   :-0.483270   Min.   :0.06617   Min.   :4.250   Min.   :4.347   Min.   :  81.7   Min.   : 44.58   Min.   : 132.4
 1st Qu.:-0.168765   1st Qu.:0.07359   1st Qu.:5.092   1st Qu.:5.109   1st Qu.: 172.1   1st Qu.:104.81   1st Qu.: 262.3
 Median :-0.007466   Median :0.07702   Median :5.623   Median :5.504   Median : 256.2   Median :155.19   Median : 391.0
 Mean   :-0.007696   Mean   :0.08041   Mean   :5.604   Mean   :5.596   Mean   : 354.6   Mean   :214.37   Mean   : 541.8
 3rd Qu.: 0.170600   3rd Qu.:0.08446   3rd Qu.:6.015   3rd Qu.:6.049   3rd Qu.: 442.2   3rd Qu.:263.39   3rd Qu.: 675.1
 Max.   : 0.499617   Max.   :0.11909   Max.   :6.718   Max.   :7.218   Max.   :1412.5   Max.   :881.10   Max.   :2123.0
> summary(zinc_KED)
Object of class SpatialPixelsDataFrame
Coordinates:
     min    max
x 178460 181540
y 329620 333740
Is projected: NA
proj4string : [NA]
Number of points: 3103
Grid attributes:
  cellcentre.offset cellsize cells.dim
x            178460       40        78
y            329620       40       104
Data attributes:
   var1.pred           var1.var          zinc            lower           upper
 Min.   :4.305   Min.   :0.06639   Min.   :  78.74   Min.   : 41.69   Min.   : 130.8
 1st Qu.:5.065   1st Qu.:0.07553   1st Qu.: 165.12   1st Qu.: 98.77   1st Qu.: 254.6
 Median :5.499   Median :0.08033   Median : 254.41   Median :153.57   Median : 388.1
 Mean   :5.612   Mean   :0.08391   Mean   : 374.48   Mean   :223.85   Mean   : 576.8
 3rd Qu.:6.102   3rd Qu.:0.08970   3rd Qu.: 466.27   3rd Qu.:277.23   3rd Qu.: 723.0
 Max.   :7.292   Max.   :0.12821   Max.   :1524.20   Max.   :935.85   Max.   :2428.3
```

Part 2: Regression kriging with random forests

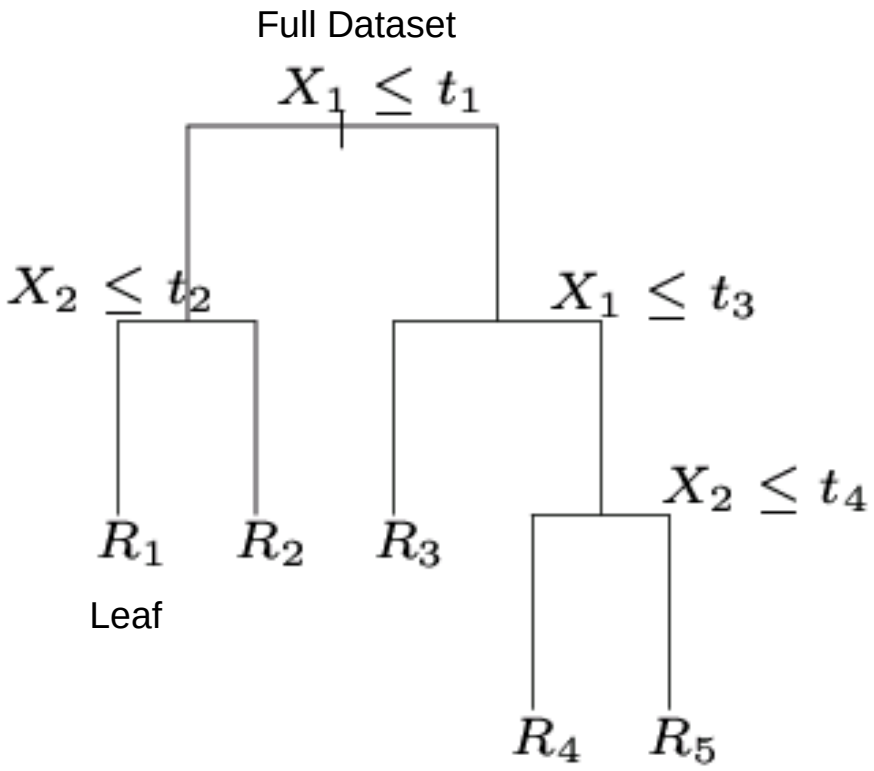
ISRIC World Soil Information

Classification and regression trees (CART)

- Classification tree for categorical data; Regression tree for continuous data
- Overcome limitations of classical (linear) model:
  - non-linear relationships;
  - *n* of covariates > *n* observations;
  - interactions of categorical covariates that result in sparse cell counts;
  - non-parametric;
  - can handle missing values
- Recursive partitioning of the data based on binary splitting of the data using covariates

Growing a tree

- Evaluate all covariates for each split
- Split is chosen so that a maximum reduction of the error is achieved. Each node is more pure than its parent node.
- Splitting process is repeated for next two nodes, etc.
- Greedy algorithm
- The prediction at a leaf is the **mean** value of the data points (RT) or the **modal** class (CT)

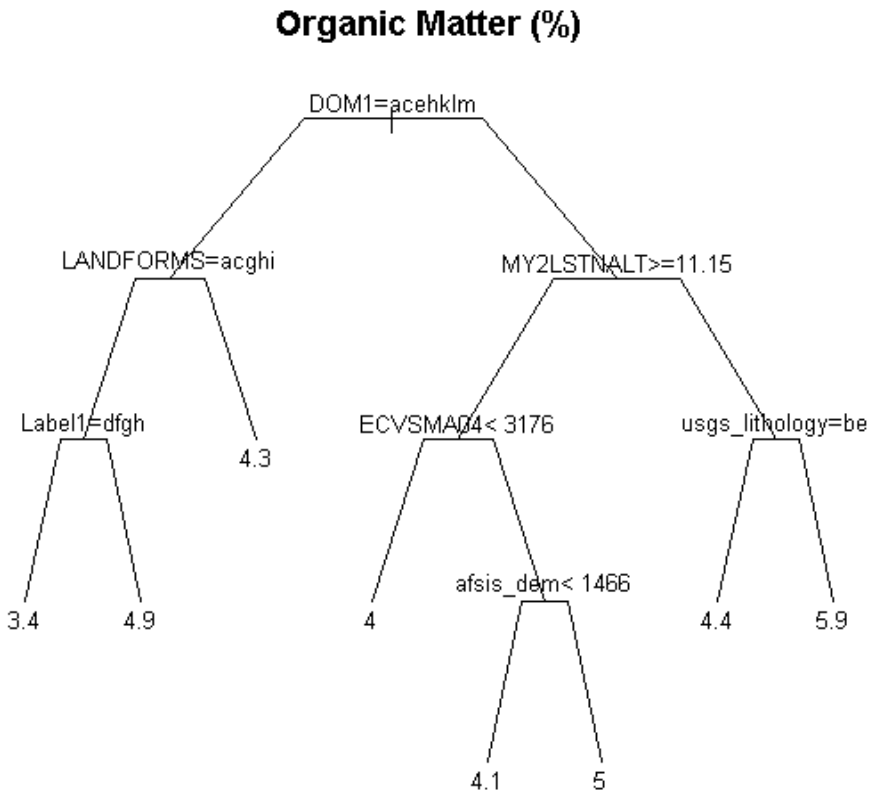


Source: Hastie et al., 2009. The Elements of Statistical Learning

Growing a tree in R

- rpart, tree, party packages
- Fitting a tree model to soil organic matter content

```
# fit regression tree
rpart.prune <- rpart(
  trend,
  data = samples,
  method = "anova", # use 'class' for categorical data
  control = rpart.control(
    minsplit = 20,  # default is 20
    minbucket = 10, # minimum observations in terminal
    cp = 0.018,     # cost-complexity parameter, used f
    xval = 10       # number of cross-validations; defa
  )
)
```



Random Forests

- Limitations of CART:
  - Trees are known to be instable: sensitive to small changes in learning data -> tree structure can be completely altered. Predictions of single trees show high variability.
  - Danger of over-fitting.
- Can be avoided using **ensemble methods**: base prediction on a whole set of trees rather than a single tree
- Ensemble methods use the fact that trees are unstable but on average produce the right result.
- Random forests is such ensemble method: a forest of trees is grown; the prediction is an aggregation of the individual tree predictions.

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

How does it work?

- Random forests combine bootstrap aggregation ('bagging') with random selection of predictors.

- Algorithm:
  - Draw a bootstrap sample:
    - random selection of 2/3 of the training data; repeat *n* times.
  - Grow an unpruned tree to each bootstrap sample
    - random predictor selection: for each split in each tree a random subset is selected from the predictor variables. The best split is chosen from among the selected predictors.
  - Predict new data by aggregating the predictions of the *n* trees.
    - Average for continuous variables
    - Majority vote for categorical variables.

Out-Of-Bag (OOB) accuracy assessment

- Random Forest comes with an internal accuracy assessment (based on cross-validation; no need to do a separate assessment).
- The algorithm sets aside 1/3 of the training data for each tree grown (**out-of-bag data**).
- OOB data can be used to asses prediction accuracy:
  - predict the data not in the bootstrap sample for each tree
  - Aggregate the OOB predictions (each data point will be OOB ~36% of the times): mean (continuous data), majority vote (categorical data).

Random Forests in R

- randomForest package (party; cforest)

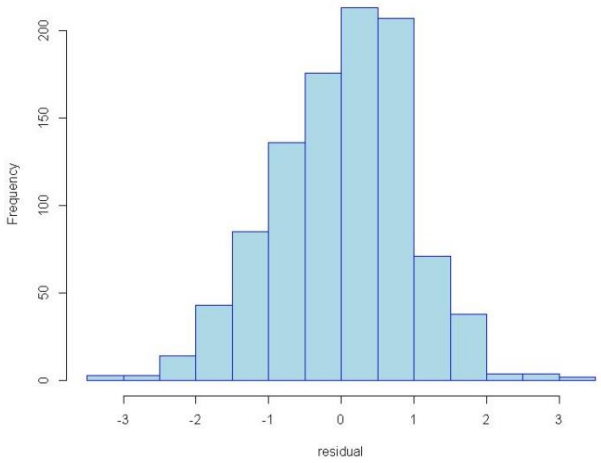
```
### Random forests modelling ###
library(randomForest)

# create object with target variable
tval <- samples$om

#create object with covariates
covar <- samples[,20:157]

# fit random forests model
rf <- randomForest(
  x = covar,
  y = tval,
  mtry = 15,          # number of randomly selected covariates, default p/3, sqrt(p)
  ntree = 1000,       # number of trees
  nodesize = 10,      # minimum size of terminal nodes
  importance = TRUE,  # assess importance of predictors
  keep.forest = TRUE, # keep the forest in the output
  keep.inbag = TRUE   # keep track of which samples are "in-bag"
)
```

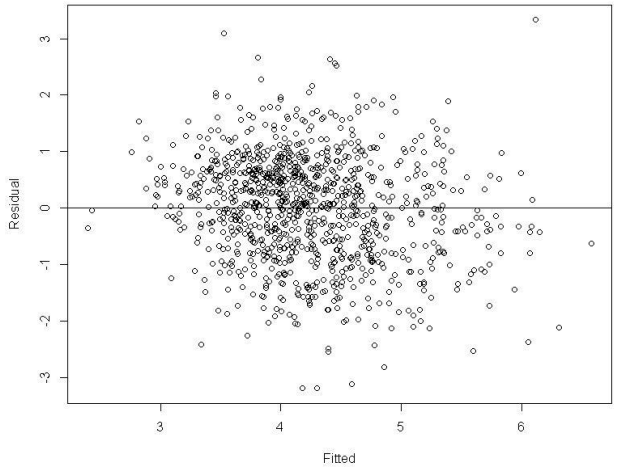
R output

```
> str(rf,2)
List of 17
 $ call           : language randomForest(x = covar, y = samples$om, ntree = 1000, mtry = 15,
E, keep.forest = TRUE,      keep.inbag = TRUE)
 $ type           : chr "regression"
 $ predicted      : Named num [1:999] 4.09 5.02 5.11 3.7 4.66 ...
  ..- attr(*, "names")= chr [1:999] "1" "2" "3" "4" ...
 $ mse            : num [1:1000] 1.51 1.44 1.48 1.34 1.34 ...
 $ rsq            : num [1:1000] -0.01347 0.03017 0.00113 0.09671 0.09927 ...
 $ oob.times      : int [1:999] 349 367 322 368 343 372 339 376 352 341 ...
 $ importance     : num [1:135, 1:2] 0.00498 0.00608 0.09721 0.00844 0.06047 ...
  ..- attr(*, "dimnames")=List of 2
 $ importanceSD   : Named num [1:135] 0.000825 0.000994 0.004444 0.001089 0.003892 ...
  ..- attr(*, "names")= chr [1:135] "Label1" "LITH_DESC" "DOM1" "DOMSOILS" ...
 $ localImportance: NULL
 $ proximity      : NULL
 $ ntree          : num 1000
 $ mtry           : num 15
 $ forest         :List of 11
  ..$ ndbigtree    : int [1:1000] 333 333 333 333 333 333 333 333 333 333 ...
  ..$ nodestatus   : int [1:333, 1:1000] -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 ...
  ..$ leftDaughter : int [1:333, 1:1000] 2 4 6 8 10 12 14 16 18 20 ...
  ..$ rightDaughter: int [1:333, 1:1000] 3 5 7 9 11 13 15 17 19 21 ...
  ..$ nodepred     : num [1:333, 1:1000] 4.21 4.07 4.94 3.97 5.07 ...
  ..$ bestvar      : int [1:333, 1:1000] 122 5 118 81 52 22 29 37 26 2 ...
  ..$ xbestsplit   : num [1:333, 1:1000] -213.5 1531 10.4129 0.0873 1759 ...
  ..$ ncat         : Named int [1:135] 13 3 13 9 11 9 15 9 4 6 ...
  .. ..- attr(*, "names")= chr [1:135] "Label1" "LITH_DESC" "DOM1" "DOMSOILS" ...
  ..$ nrnodes      : int 333
  ..$ ntree        : num 1000
  ..$ xlevels      :List of 135
  .. .. [list output truncated]
 $ coefs          : NULL
 $ y              : num [1:999] 4.29 5.28 4.96 3.77 4.41 1.64 3.33 5.24 4.16 4.28 ...
 $ test           : NULL
 $ inbag          : int [1:999, 1:1000] 1 0 0 1 1 1 0 0 1 1 ...
```

Random forests residuals

- randomForest returns predicted values of the input data based on out-of-bag samples.
- residual = rf$predicted – rf$y



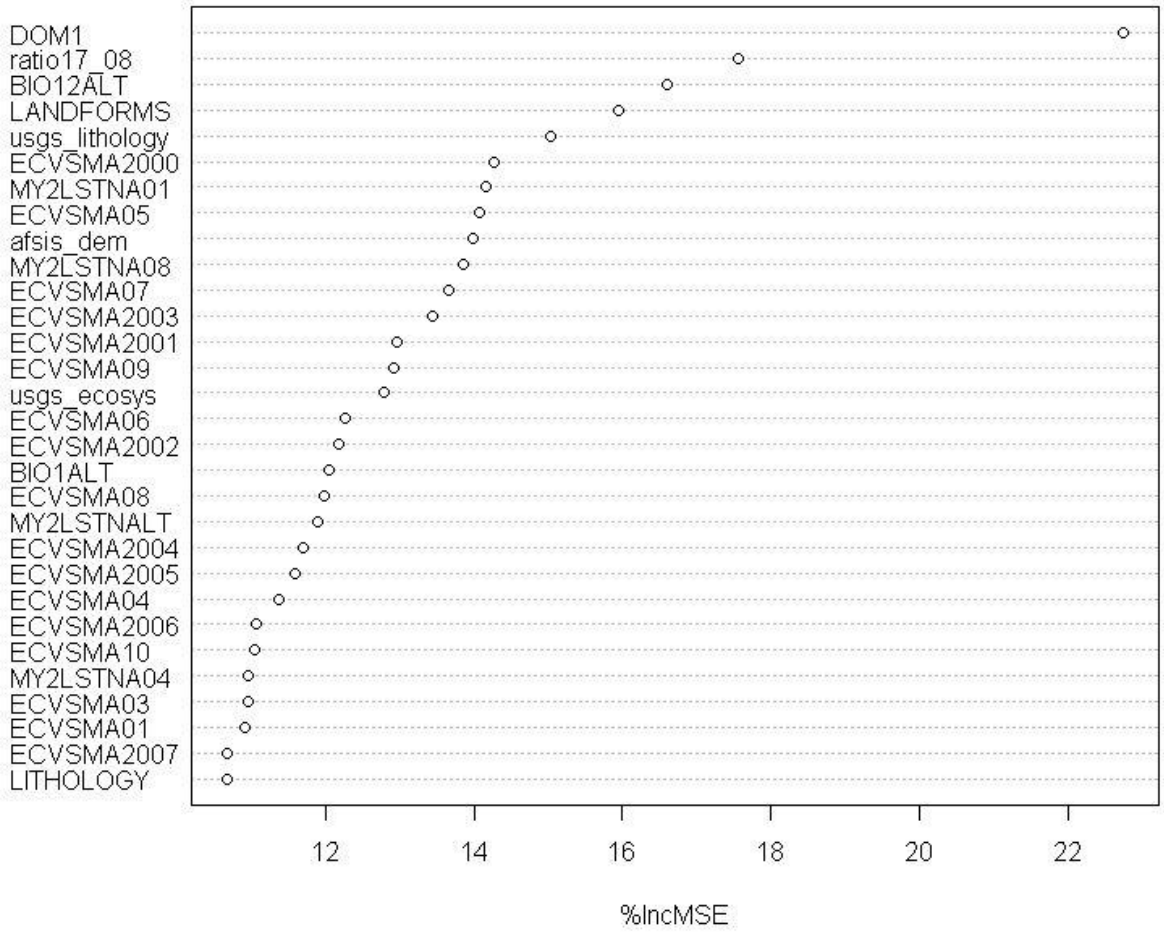


- fit variogram, krige residuals, add RF predictions

Variable importance

- Ensembles of trees are not easy to interpret: no such thing as an average tree; an individual tree does not tell much
- Ensemble can reflect the potentially (complex) effect of a variable on the response -> assess the relevance of each variable over all trees of the ensemble
- **Variable importance plot:** shows how much prediction error increases when the values of one predictor are permuted (break association with response) while all others are left unchanged
- Permuted variable is used together with other variables to predict the response -> prediction accuracy will decrease

Variable importance plot



Be aware

- No clear interpretation
- Prediction uncertainty not easy to quantify (computationally intensive)
- Spatial correlation cannot be accounted for
- Bias in variable selection (Strobl et al. 2009)
- Stability of the forest depends on ntree, mtry settings

Tree-based methods: resources

Psychological Methods
2009, Vol. 14, No. 4, 323–348

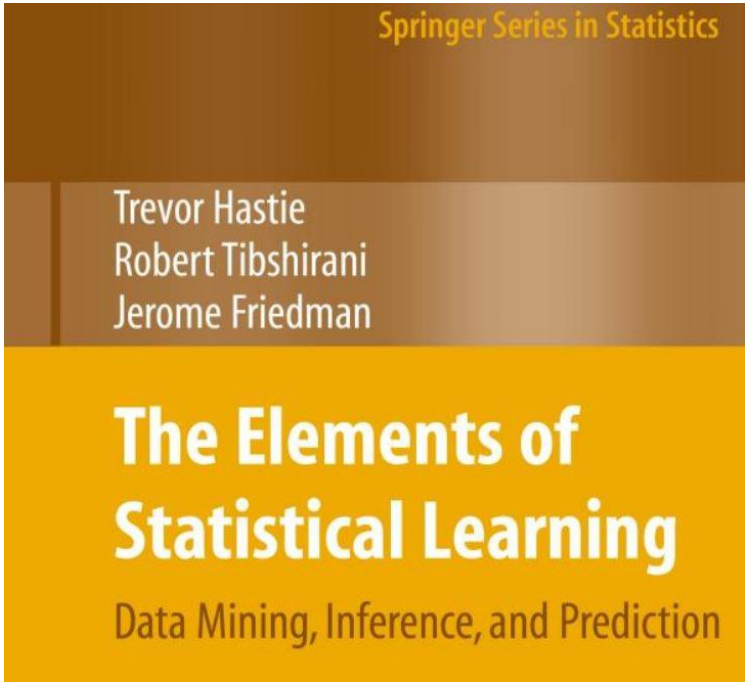
1082-989X/09/$12.00 DOI: 10.1037/a0016973

An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests

Carolin Strobl
Ludwig-Maximilians-Universität Munich

James Malley
National Institutes of Health

Gerhard Tutz
Ludwig-Maximilians-Universität Munich



Machine Learning Algorithms

- Machine Learning Algorithms for soil science data: R tutorial

And now…
let's practice

ISRIC World Soil Information