Introduction to Correlation and Regression

D G Rossiter

February 17, 2016

Copyright © 2007–2012, 2015-16 D G Rossiter
All rights reserved. Reproduction and dissemination of the work as a whole (not parts) freely permitted if this original copyright notice is included. Sale or placement on a web site where payment must be made to access this document is strictly prohibited. To adapt or translate please contact the author (dgr2@cornell.edu).

Topics

1. Correlation
2. Simple linear regression
3. Model validation
4. Structural analysis
5. Multiple linear regression
6. Regression trees and random forests
7. Factor analysis (Principal Components Analysis)
8. Robust methods

Computing environment

Output produced by R; see `http://www.r-project.org`

Topic: Relations between variables

Given a **dataset** which contains:

- **sampling units** ("records", "individuals")
- **items** measured on each sampling unit ("variables")

What is the "relation" between the variables?

- Association: **what?**
- Explanation: **why?**
- Causation: **how?**
- Prediction: **what if?**

Types of relations between variables

1. Variables are of **equal** status

   (a) A bivariate **correlation** between two variables;
   (b) A multivariate **correlation** between several variables;
   (c) A **structural relation** between two variables;
   (d) A **structural relation** between several variables (e.g. principal components).

2. Variables have **different** status

   (a) A **simple regression** of one **dependent** variable on one **independent** variable;
   (b) A **multiple regression** of one **dependent** variable on several **independent** variable.
   (c) A **hierachical model** (tree) relating a **dependent** variable to several **independent** variables.

Regression

This is a general term for **modelling** one or more:

- **response** variables (**predictands**, mathematically **dependent**), from one or more
- **predictor** variables (mathematically **independent**)

Note: The "response" and "predictor" are **mathematical** terms, *not necessarily* **"effect"** and **"cause"** – that requires **meta-statistical** reasoning.

Linear models

- All variables are related with **linear** equations.
- These are easy to work with and have good mathematical properties.
- Their **interpretation** is easy (proportional relations).
- The linear relation can be after **transformation** of one or more variables, to **linearize** the relation.
- Relations that can not be linearized are **intrinsically non-linear**.

Is the relation linear?

Reference: Anscombe, F. J. Graphs in Statistical Analysis. *American Statistician* **27**, 17-21, 1973

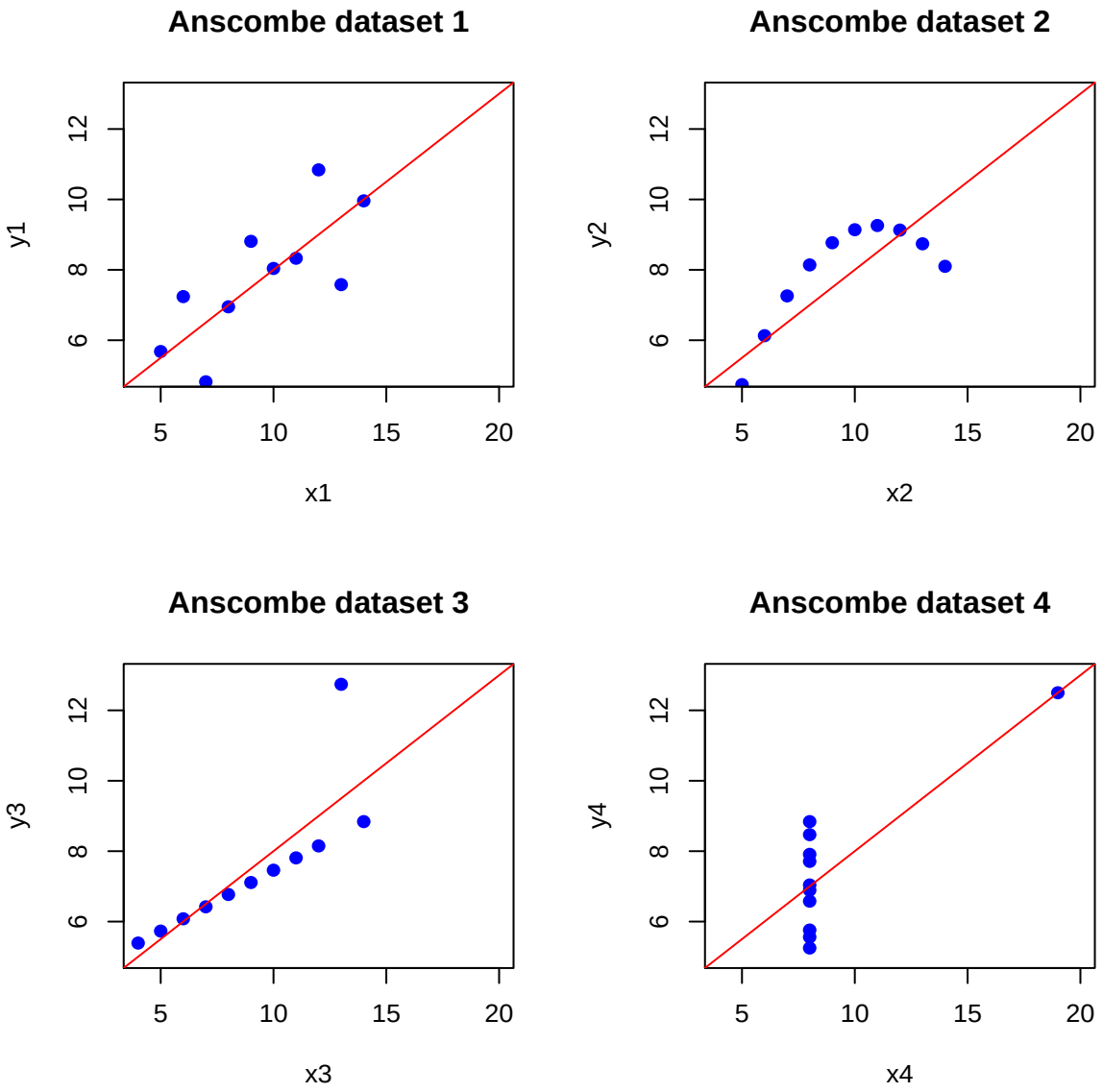
Four different bivariate datasets, all with the exact:

- same **correlation coefficient** $r = 0.81$;
- same **linear regression equation** $y = 3 + 0.5x$

Quantitatively: identical correlation and regression

Qualitatively: very different interpretations

Anscombe's quartet



Interpretation

1. noisy linear

2. perfect quadratic

3. perfect linear, one outlier (observation not fitting the pattern)

4. ?? one point is controlling the relation, no way of knowing:

   (a) variability at that value of the predictor
   (b) intermediate points

Topic: Correlation

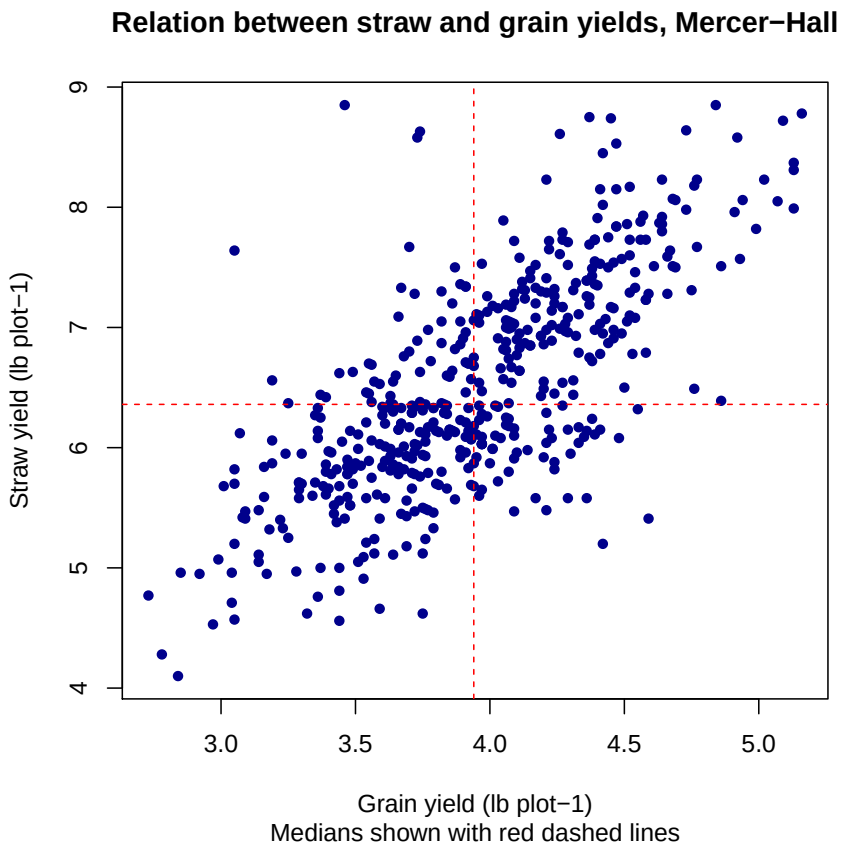
- Measures the **strength of association** between two variables measured on the same object:
  - -1 (perfect **negative** correlation)
  - 0 (no correlation)
  - $+1$ (perfect **positive** correlation).
- The two variables have logically equal status
- No concept of **causation**
- No **functional relation**, no way to **predict**

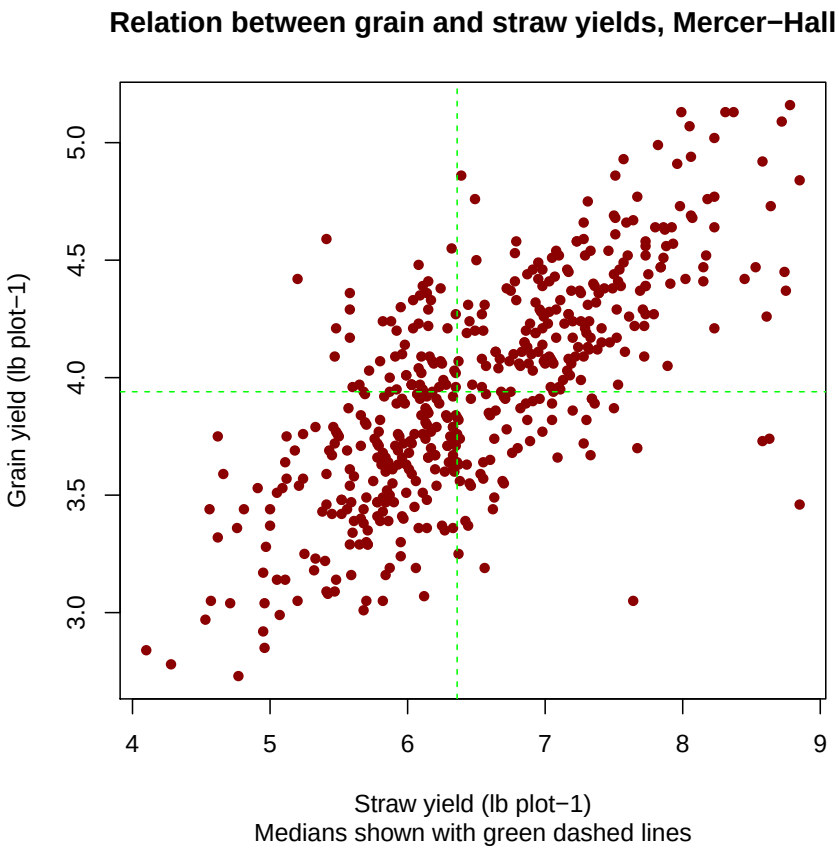
Example dataset

Source: W B Mercer and A D Hall. The experimental error of field trials. *The Journal of Agricultural Science (Cambridge)*, **4**: 107–132, 1911.

- A **uniformity** trial: 500 supposedly identical plots within one field
- All planted to one variety of **wheat** and treated identically
- Measured variables: **grain and straw** yields, lbs per plot, precision of 0.01 lb (0.00454 kg)

Bivariate scatterplot





What kind of relation between the two variables?

1. Variables are of **equal** status

   (a) A bivariate **linear correlation** between the two variables (straw and grain yields);
   (b) A **linear structural relation** between the two yields.

2. Variables have **different** status

   (a) A univariate **linear regression** of straw (dependent) on grain (independent) yield;
   (b) A univariate **linear regression** of grain (dependent) on straw (independent) yield.

We begin with **linear correlation**.

Measuring correlation

1. **Parametric**:
   - Assumes some bivariate distribution
   - e.g. Pearson's product moment correlation coefficient (PMCC) r;
2. **Nonparametric**
   - Uses ranks, not distributions
   - e.g. Spearman's ρ.

Measuring the strength of a bivariate relation

- The **theoretical covariance** of two variables X and Y

$$\begin{aligned} \mathrm{Cov}(X,Y) &= E\{(X-\mu_X)(Y-\mu_Y)\} \\ &= \sigma_{XY} \end{aligned}$$

- The **theoretical correlation coefficient**: covariance normalized by population standard deviations; range $[-1 \ldots 1]$:

$$\begin{aligned} \rho_{XY} &= \frac{\mathrm{Cov}(XY)}{\sigma_X \cdot \sigma_Y} \\ &= \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y} \end{aligned}$$

Sample vs. population covariance and correlation

- Sample $\overline{x} = 1/n \sum x_i$ estimates population μ_X
- Sample $s_x = \sqrt{\frac{1}{n-1} \sum (x_i - \overline{x})^2}$ estimates population σ_X
- Sample $s_{xy} = \frac{1}{n-1} \sum_{i=1} (x_i - \overline{x}) \cdot (y_i - \overline{y})$ estimates population σ_{XY}
- Sample $r_{xy} = \frac{s_{xy}}{s_x \cdot s_y}$ estimates population ρ_{XY}

Covariance vs. correlation

Covariance: in original units, original scale:

E.g. mean grain, straw yields in lbs per plot, and their covariance in (lbs per plot)^2

```
[1] "means: Grain: 3.949 ; Straw: 6.515"
```

```
[1] "standard deviations: Grain: 0.458 ; Straw: 0.898"
```

```
[1] "Covariance: 0.3004"
```

Correlation: standardized to a $(-1 \ldots +1)$ scale:

Both variables: subtract **mean** and divide by **standard deviation**:

```
[1] "Correlation: 0.7298"
```

Assumptions for parametric correlation

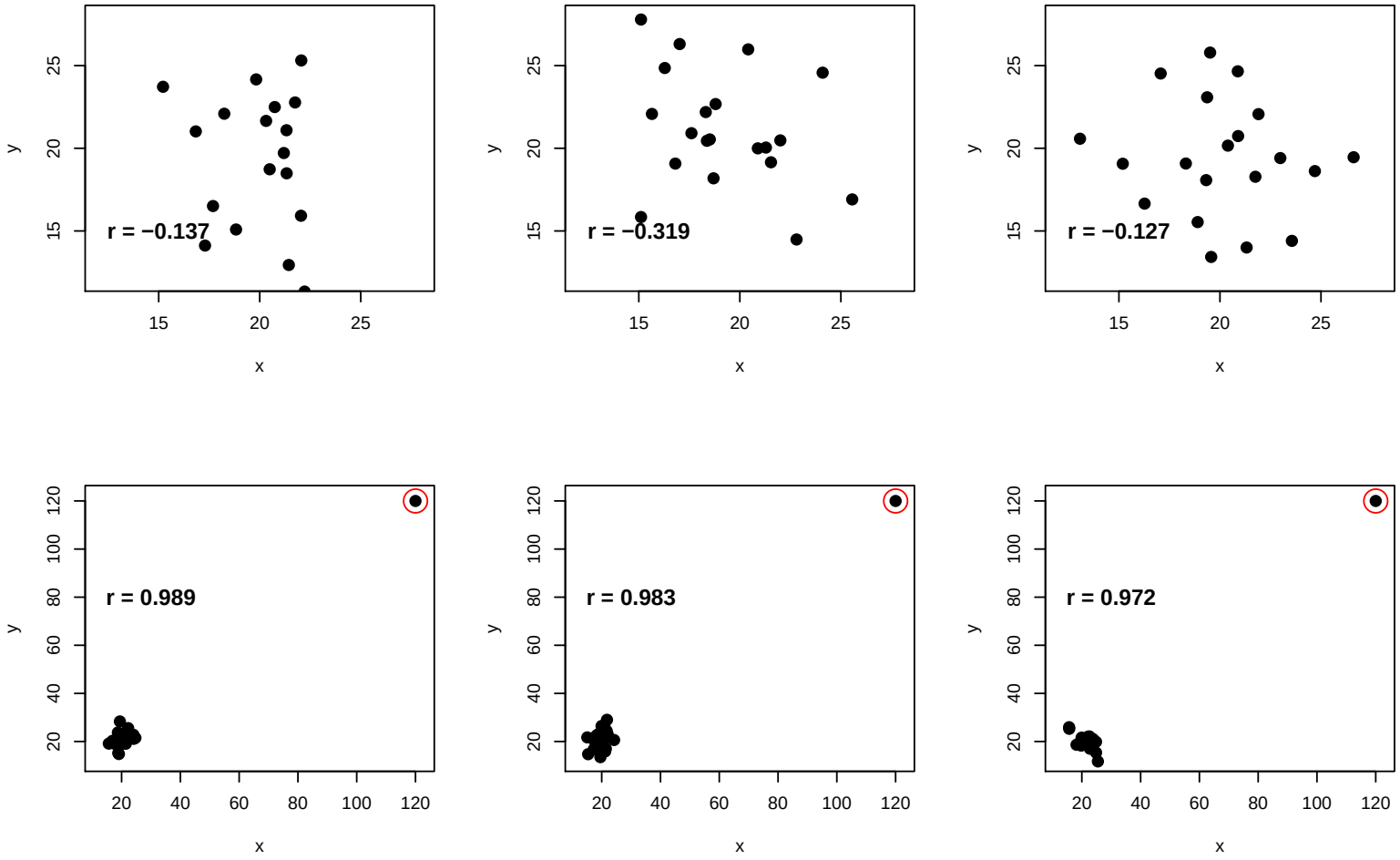
Requires **bivariate normality**; do these two variables meet that?

If the assumption isn't met, must use either:

- **transformations** to bivariate normality (may be impossible), or
- **ranks** (see below)

Clear violation of assumptions

One point can **arbitrarily** change the correlation coefficient Example: 3 **uncorrelated** random samples (theoretical $\rho = 0$), without/with one **contaminating** observation:



Visualizing bivariate normality

To **visualize** whether a particular sample meets the assumption:

1. Draw **random samples** that in theory **could** have been observed from samples of the same size, **if** the data are from the theoretical **bivariate normal** distribution required for PPMC. This is **simulating** a sample from known (assumed) population.

   Note: R functions for simulating samples:

   - `rnorm` (univariate normal);
   - `mvrnorm` from the `MASS` package (multivariate normal)

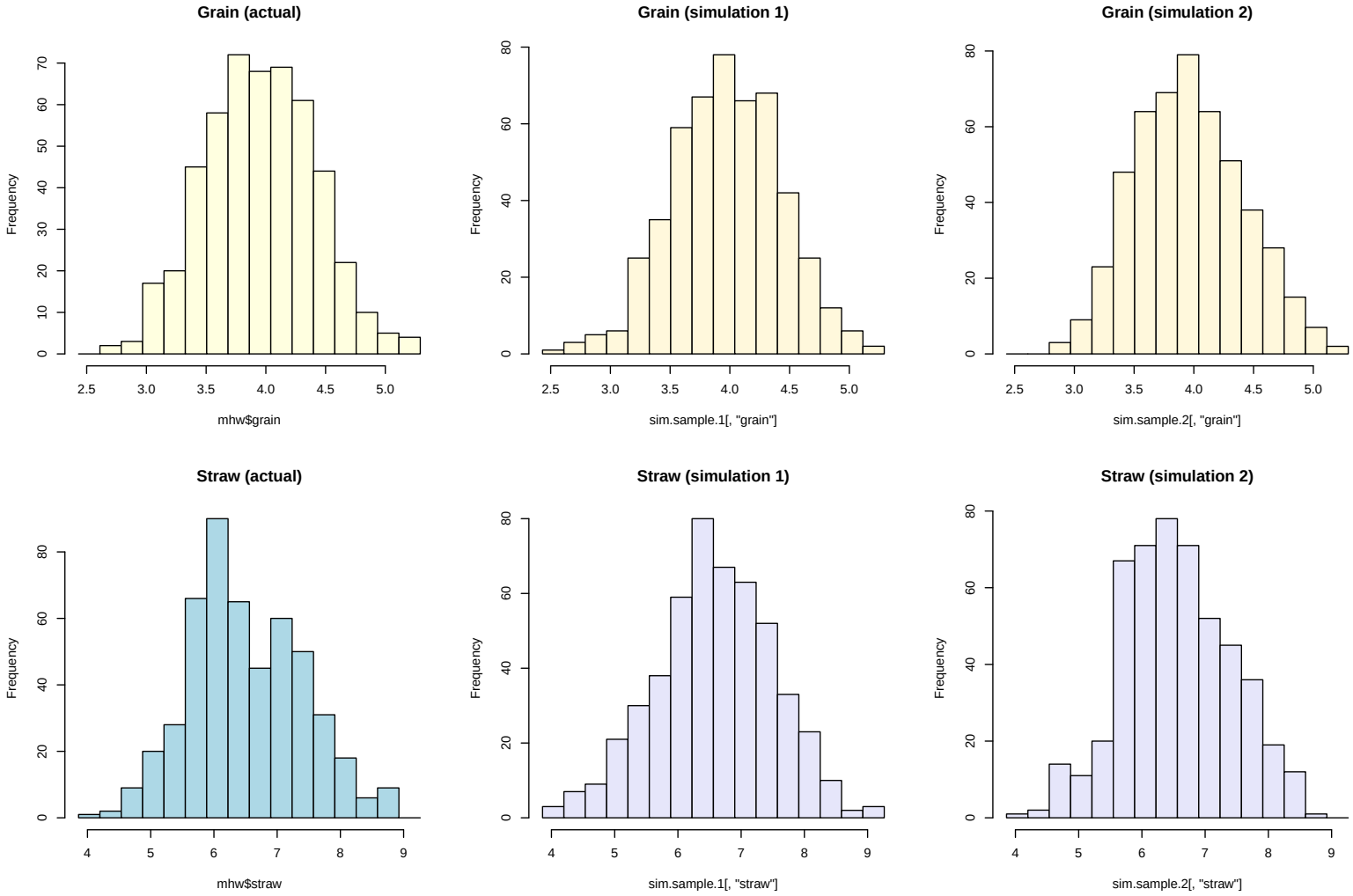
2. Display them next to the **actual sample**:

   (a) **univariate**: histograms, Q-Q plots
   (b) **bivariate**: scatterplots

They should have the same form.

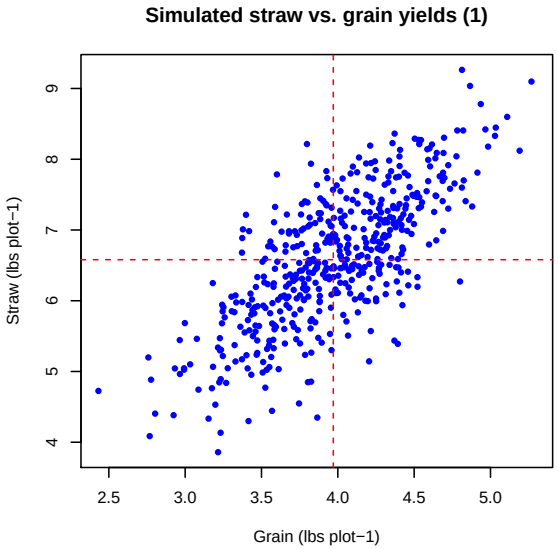
Histograms – simulated vs. actual

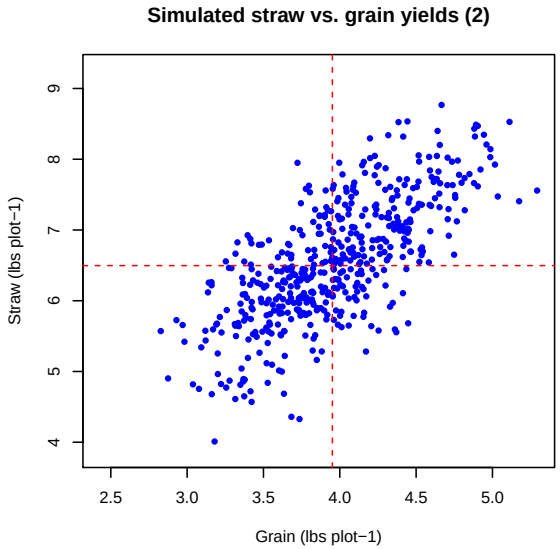
Do the single variables each appear to be normally-distributed?

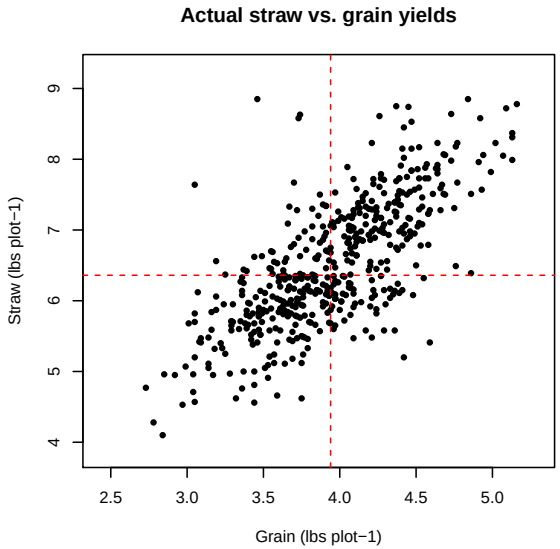


Scatterplots – simulated vs. actual

Do the two variables together appear to be normally-distributed?







Values vs. ranks

Non-parametric methods compute the parametric coefficient on **ranks**:

Lowest-yielding grain and straw plots:

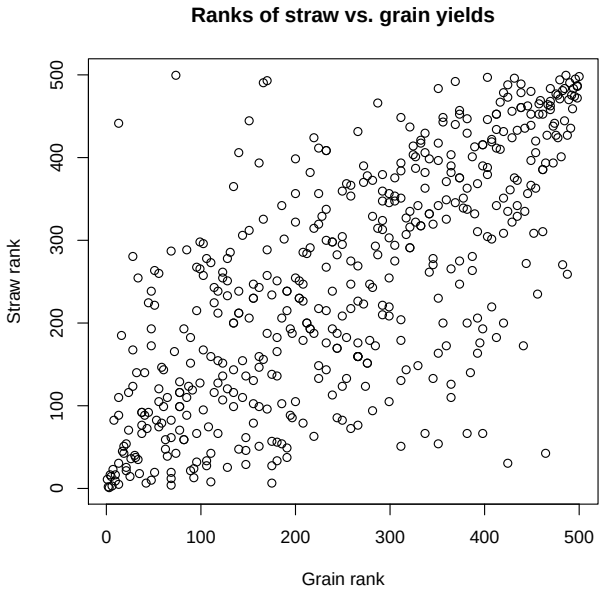
```
 [1] 338 467 470 339 336 441 149 319  81 228 164 273
```

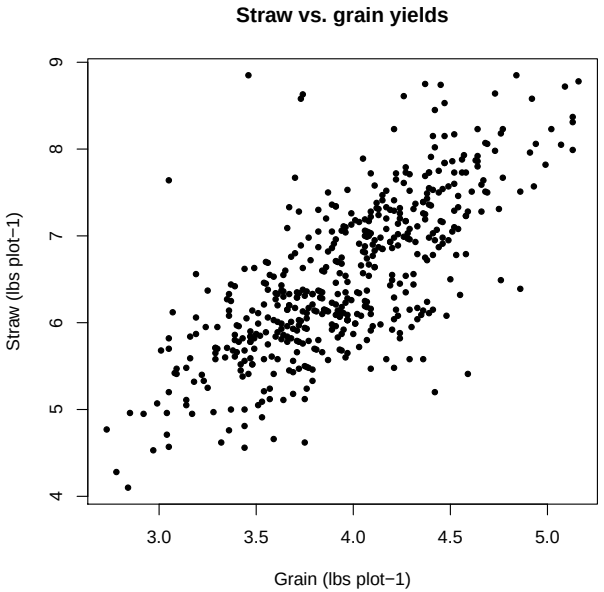
```
 [1] 470 467 441 447 427 284 444 460  81 401 338 469
```

Some plots with their ranks and yields:

```
  grain straw rank(mhw$grain) rank(mhw$straw)
1  3.63  6.37           123.0           254.5
2  4.07  6.24           299.0           219.5
3  4.51  7.05           445.5           356.5
4  3.90  6.91           228.0           329.0
5  3.63  5.93           123.0           136.0
6  3.16  5.59            23.5            70.5
7  3.18  5.32            26.0            36.0
8  3.42  5.52            62.5            59.0
```

Scatterplots: values and ranks





Ranks always **lose information** but are **distribution-free**.

So, non-parametric correlations are usually **lower** (less powerful) – *if* the assumptions are met!

Correlation coefficients

Both computed with R function `cor`:

```
[1] "Parametric (PPMC), using method='pearson' 0.7298"

[1] "Non-parametric (Spearman), using method='spearman' 0.7196"
```

Can compute a **confidence interval** for the parametric coefficient (R function `cor.test`)

```
        Pearson's product-moment correlation

data:  mhw$grain and mhw$straw
t = 23.821, df = 498, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.68599 0.76830
sample estimates:
    cor
0.72978
```

Topic: Simple Linear Regression

Recall: **regression** is a general term for **modelling** one or more:

- **response** variables (**predictands**), from one or more
- **predictor** variables

The simplest case is **simple linear regression**:

1. One continous **predictor**
2. One continous **predictand**

Fixed effects model

$$Y_i = BX_i + \varepsilon_i$$

All error ε is associated with the **predictand** Y_i

There is no error in the **predictors** X_i, either because:

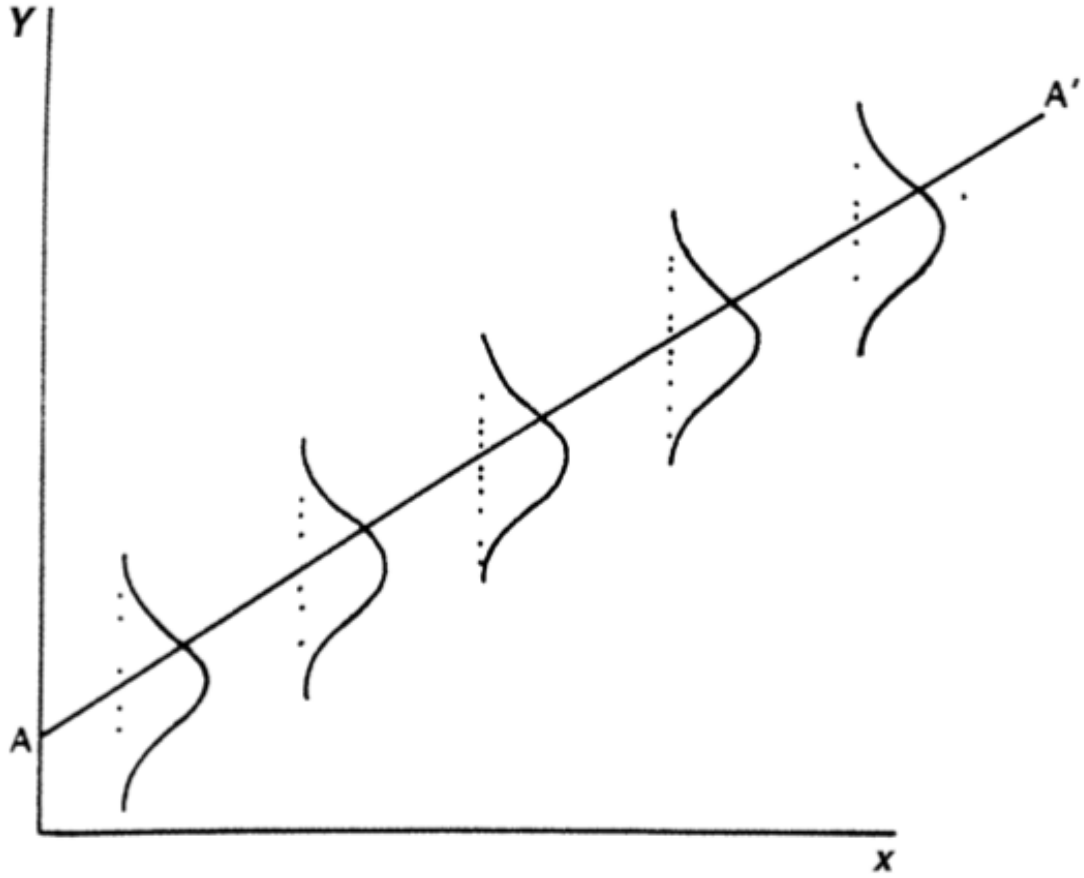
- **imposed** by researcher without appreciable error (e.g. treatments);
- **measured** without appreciable error;
- **ignored** to get "best" prediction of Y.

The **coefficients** B are chosen to **minimize** the error in the **predictand** Y.

Simplest case: a **line**: slope β_1, intercept β_0:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Fixed effects line



Source: Webster, *European Journal of Soil Science* **48**:558 (1997), Fig. 2

Least-squares solution

Two parameters must be estimated from the data:

The **slope** $\hat{\beta}_{Y.x}$ is estimated from the sample covariance s_{XY} and variances **of the predictand** s_x^2:

- $\hat{\beta}_{Y.x} = s_{XY}/s_x^2$

The **intercept** $\hat{\alpha}_{Y.x}$ is then adjusted to make the line go through the centroid $(\bar{x}, \bar{y})$:

- $\hat{\alpha}_{Y.x} = \bar{y} - \hat{\beta}_{Y.x}\bar{x}$

Note: only s_x^2 is used to compute the slope! It is a **one-way** relation, because all the error is assumed to be in the predictand.

This is the simplest case of the **orthogonal projection** (see below).

This solution has some strong **assumptions**, see below.

Matrix formulation

The general form of the linear model is $Y = XB + \varepsilon$; if there is only one response variable, this is $y = Xb + \varepsilon$.

X is called the **design matrix**, with one column per predictor, with that predictor's value for the observation i.

In the simple linear regression case, there is only one predictor variable x, and the design matrix X has an inital column of 1's (representing the mean) and a second column of the predictor variable's values at each observation:

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ & \dots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_n \end{bmatrix}$$

where the ε are **identically and indepenently distributed** (IID).

Solution by orthogonal projection

Gauss-Markov theorem: under the **assumptions** (1) linear relation; (2) errors have expectation zero; (3) errors are uncorrelated; (4) errors have equal variances:

Then: the "best linear unbiased estimator" (**BLUE**) $\hat{\mathbf{B}}$ of the regression coefficients is given by the **orthogonal projection**:

$$\hat{\mathbf{B}} = [\mathbf{X}'\mathbf{X}]^{-1}[\mathbf{X}'\mathbf{y}]$$

where $'$ indicates transposition and $^{-1}$ matrix inversion.

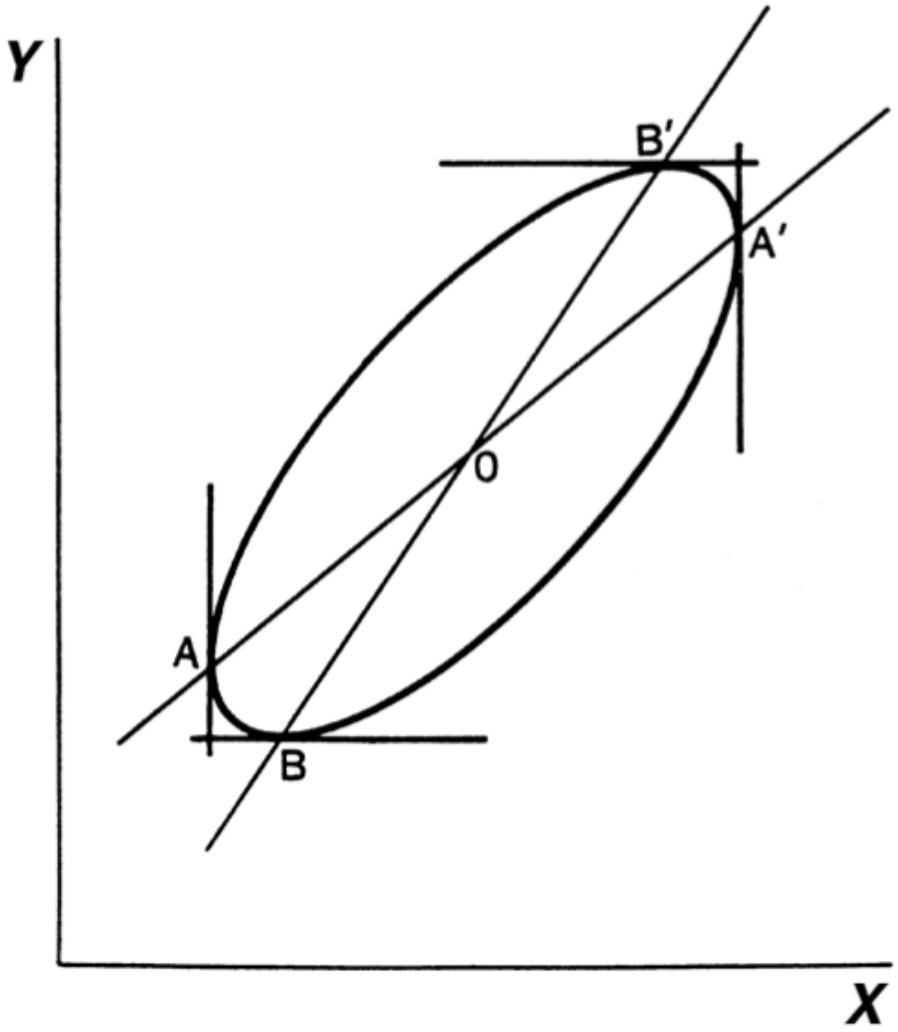
Random effects model

Error in both **predictand** y_i and **predictors** X_i.

Both variables should have Gaussian error, with some correlation. This is modelled as a **bivariate normal distribution** of two random variables, X and Y

$$\begin{aligned} X &\sim \mathcal{N}(\mu_X, \sigma_X) \\ Y &\sim \mathcal{N}(\mu_Y, \sigma_Y) \\ \rho_{XY} &= \mathrm{Cov}(X, Y)/\sigma_X\sigma_Y \end{aligned}$$

Random effects lines



Source: Webster, *European Journal of Soil Science* **48**:558 (1997), Fig. 1

Fitting a regression line

Fit a line that "best" describes the response-predictor relation.

Different levels of assumptions about functional form:

1. Exploratory, non-parametric
2. Parametric
3. Robust

A parametric linear fit

Model straw yield as function of grain yield, by **minimizing** the sum-of-squares of the **residuals** (Gaussian least-squares).

Although there is error in both the grain and straw yield (**random effects** model), the aim is to minimize error in the **predictand**.

This is because the model is used to **explain** the predictand in terms of the **predictor**, and eventually to **predict** in that direction.

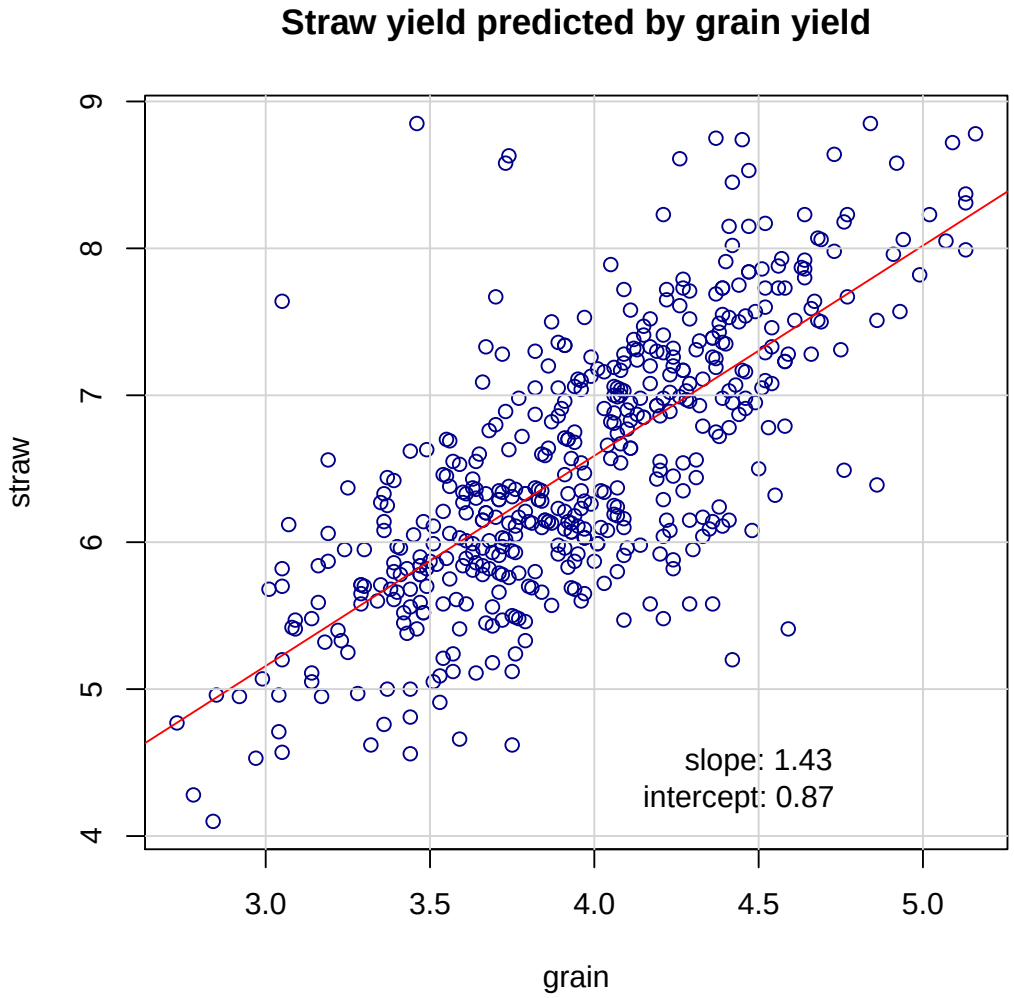
Once one variable is selected as the **response**, then the aim is to minimize that error, and the one-way **least-squares** fit is applied.

Model summary from R `lm` "linear models" fit

```
Call:
lm(formula = straw ~ grain, data = mhw)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0223 -0.3529  0.0104  0.3734  3.0342

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8663     0.2387    3.63  0.00031
grain         1.4305     0.0601   23.82  < 2e-16

Residual standard error: 0.615 on 498 degrees of freedom
Multiple R-squared: 0.533,        Adjusted R-squared: 0.532
F-statistic:  567 on 1 and 498 DF,  p-value: <2e-16
```

The summary shows **residuals** (lack of fit), **model coefficients proportion of variation** explained by model (`Adjusted R-squared`), and **probability** that rejecting various null hypotheses would be an error.

Scatterplot with best-fit line



Best-fit line: `straw` = 0.87 + 1.43 * `grain`

Assumptions of the linear model

The least-squares (parametric) solution is only valid under a strong **assumption**:

The **residuals** are **identically and indepenently distributed** (IID) from a **normal** distribution

This implies:

1. no dependence of residual on fitted values;
2. no difference in **spread** of residuals through fitted value range: **homoscedascity**
3. residuals have a **normal** distribution ($\mu_\varepsilon \equiv 0$)

Model diagnostics

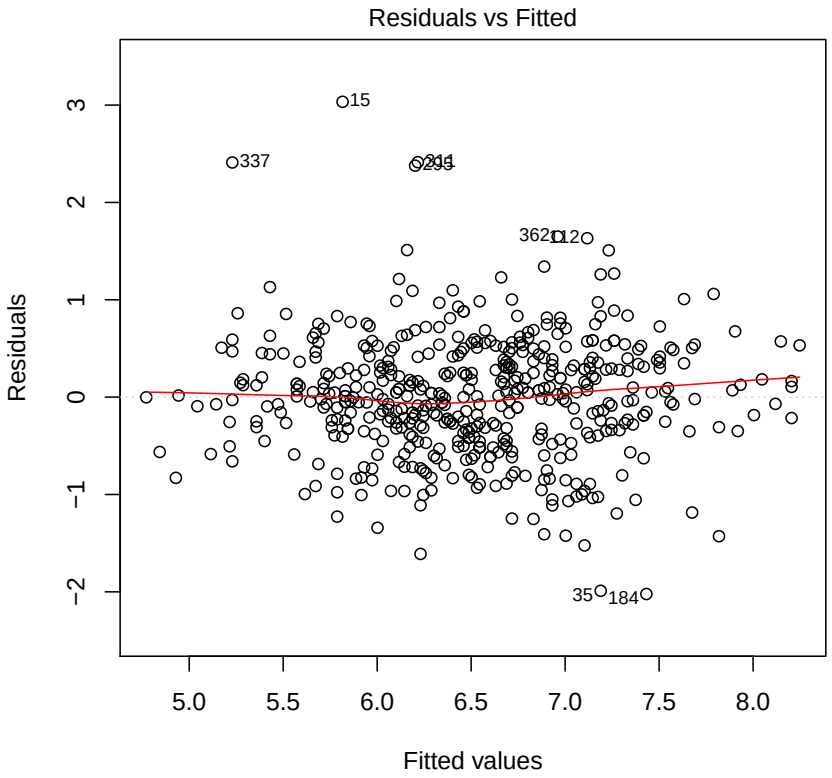
The assumptions can **visualized** and **tested**.

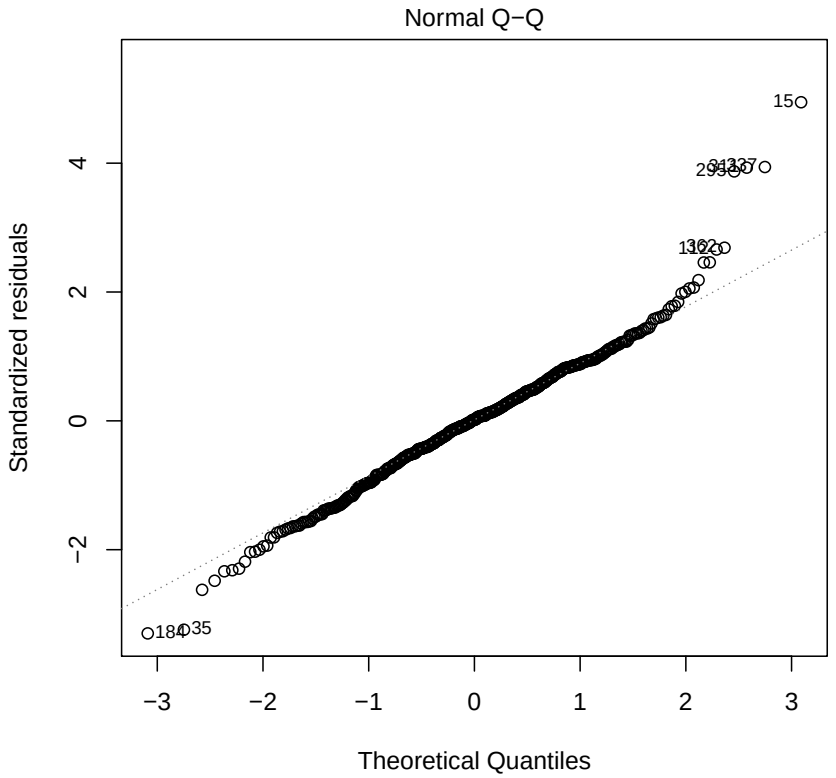
The most important tools are the **diagnostic plots**.

These are of several kinds; the most important are:

- **Normal probability plot** of the residuals
- Plot of **residuals vs. fits**
- **Leverage** of each observation (influence on fit)
- **Cook's distance** to find poorly-fitted observations

Diagnostic plots

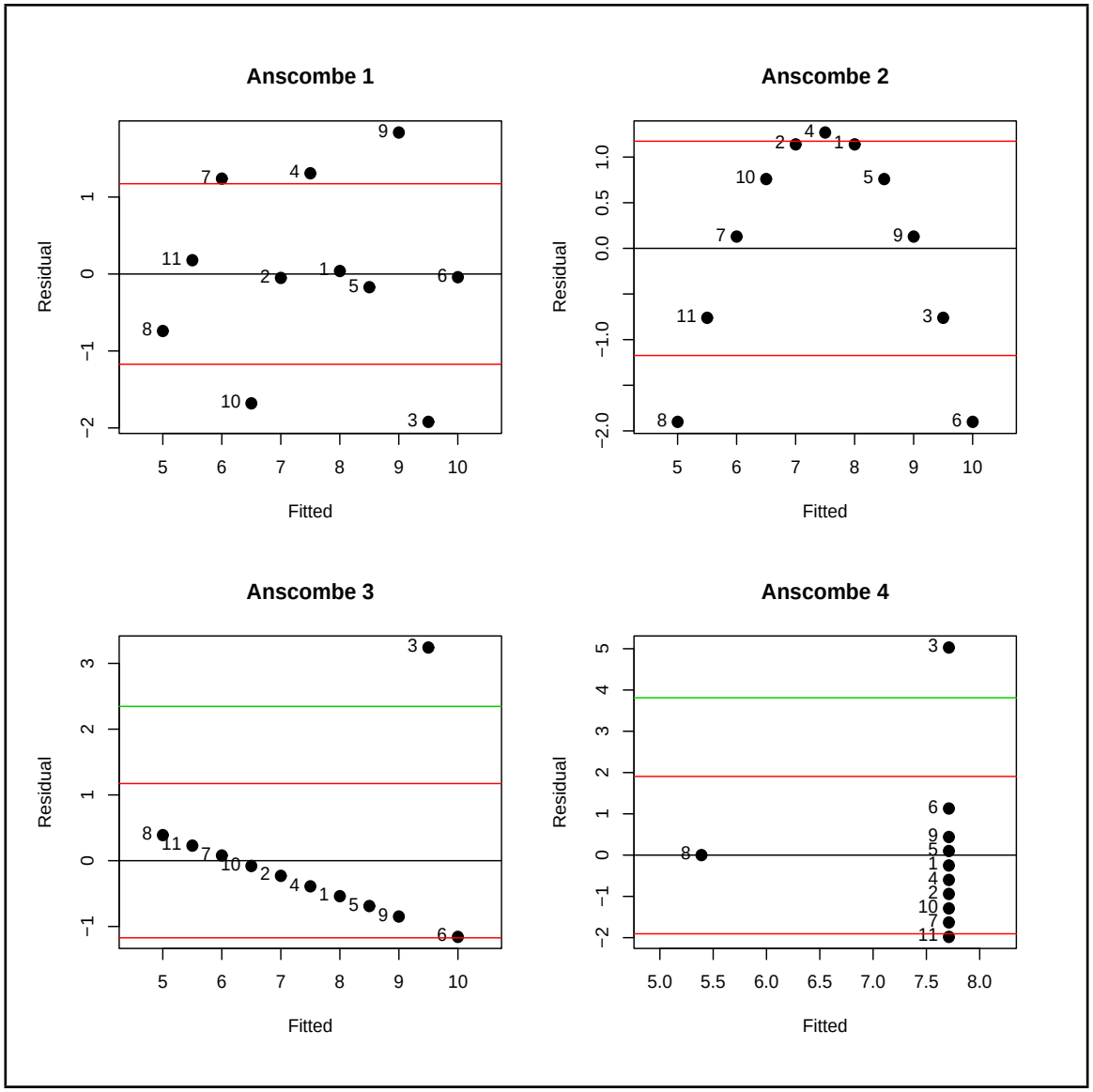




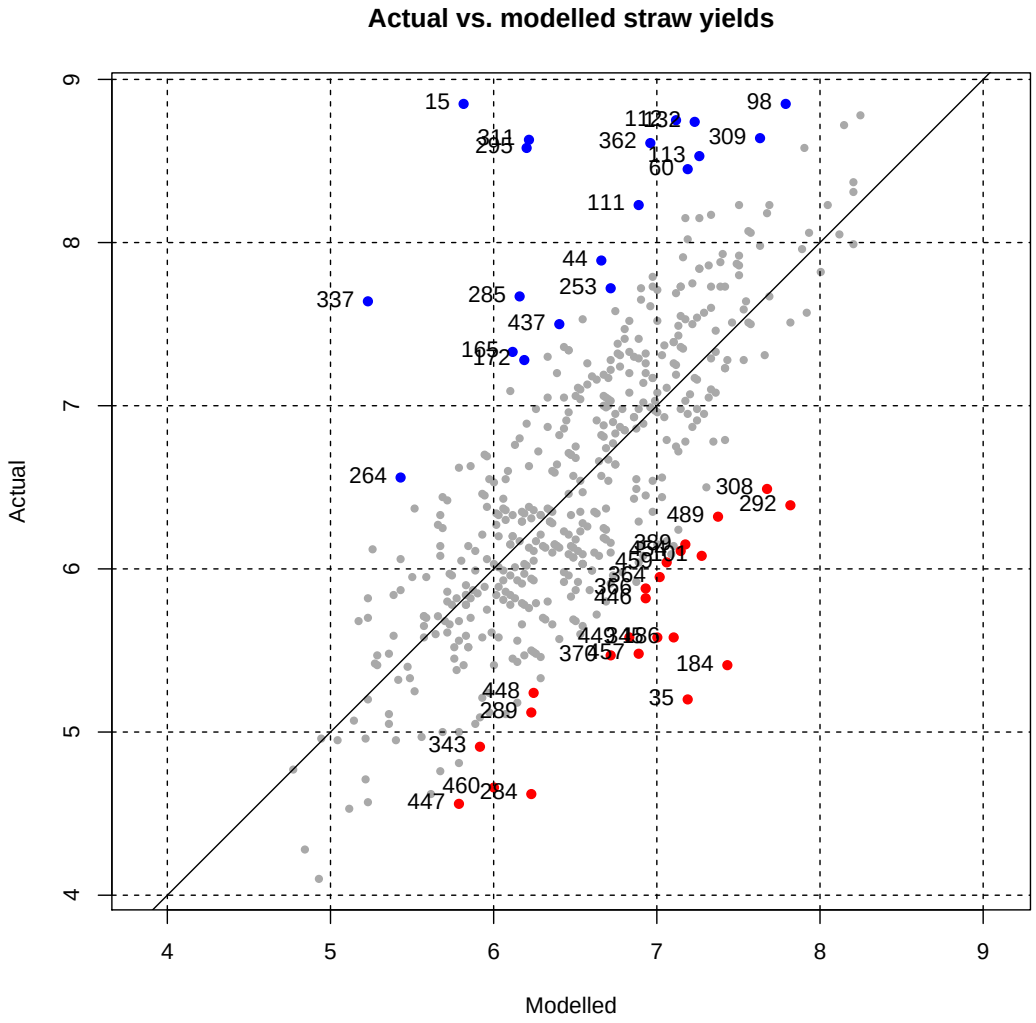
Here, a few badly **under-**fit plots, i.e., (actual - predicted) too positive.

Both tails of the Q-Q plot are too "heavy" – a **contaminated** normal distribution?

Anscombe relations: fits vs. residuals



Evaluation of model fit (1): 1-1 line



Points "should" be on 1:1 line; highlighted observations absolute residual > 1 lb. plot^{-1}.

Evaluation of model fit (2): coefficient of determination

The R^2 reported by the model summary is the **coefficient of determination**:

This is the complement of the:

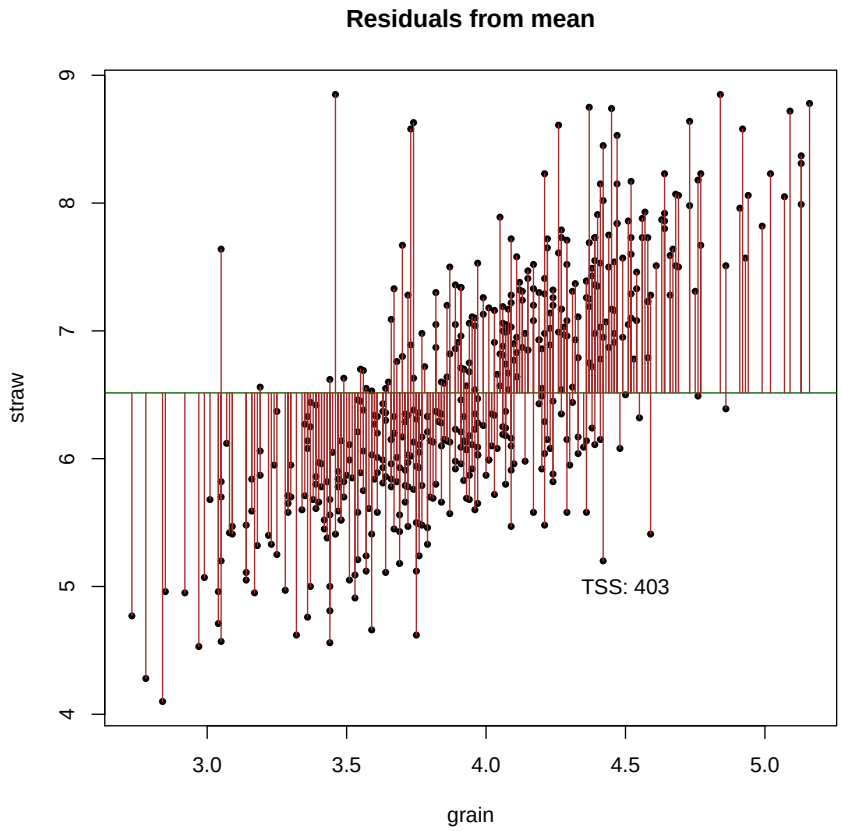
- **residual sum of squares** $RSS = \sum_{i=1}^{n} (z_i - \hat{z}_i)^2$
- . . . as a proportion of the . . .
- **total sum of squares** $TSS = \sum_{i=1}^{n} (z_i - \bar{z})^2$:

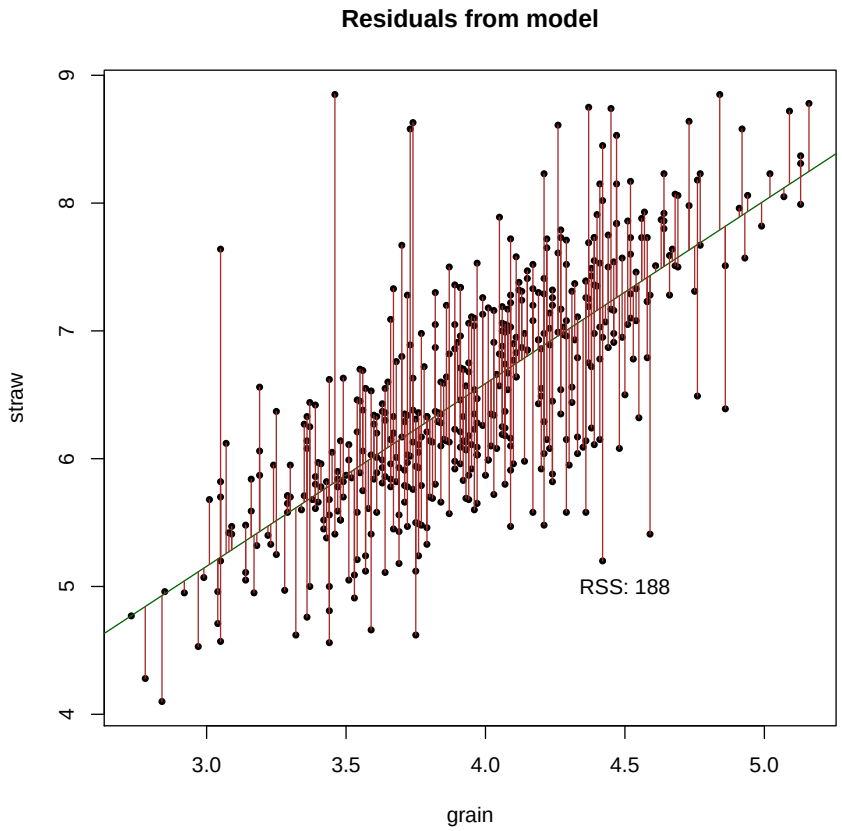
where $\hat{z}_i$ is the predicted (modelled) value and $\bar{z}$ is the mean response. So:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

$R^2 \in [0 \ldots 1]$, it measures the **proportion of variance** in the **response** (predictand) **explained** by the model, compared to the **null** model (prediction by the mean of the response).

Visualization of the coefficient of determination





Total **length of residual lines** is much shorter to the model line than to the mean line.

Calibration vs. validation

Goodness-of-fit only measures the success of **calibration** to the particular **sample** dataset.

We are actually interested in **validation** of the model over the whole **population**

- **sample** vs. **population**: representativeness, sample size

Confidence intervals of estimation

The **parameters** of the regression **equation** have some **uncertainty**, expressed as their **standard errors of estimation**:

Example: coefficients of the straw vs. grain linear regression:

```
             Estimate Std. Error
(Intercept)  0.86628   0.238715
grain        1.43050   0.060053
```

These can be multiplied by the appropriate t-value to obtain confidence intervals.

Estimation variance

Problem: the reported variance of the slope parameter $s^2_{Y.x}$ is only valid at the **centroid** of the regression, $\bar{x}$.

This variance is computed from the deviations of actual and estimated values:

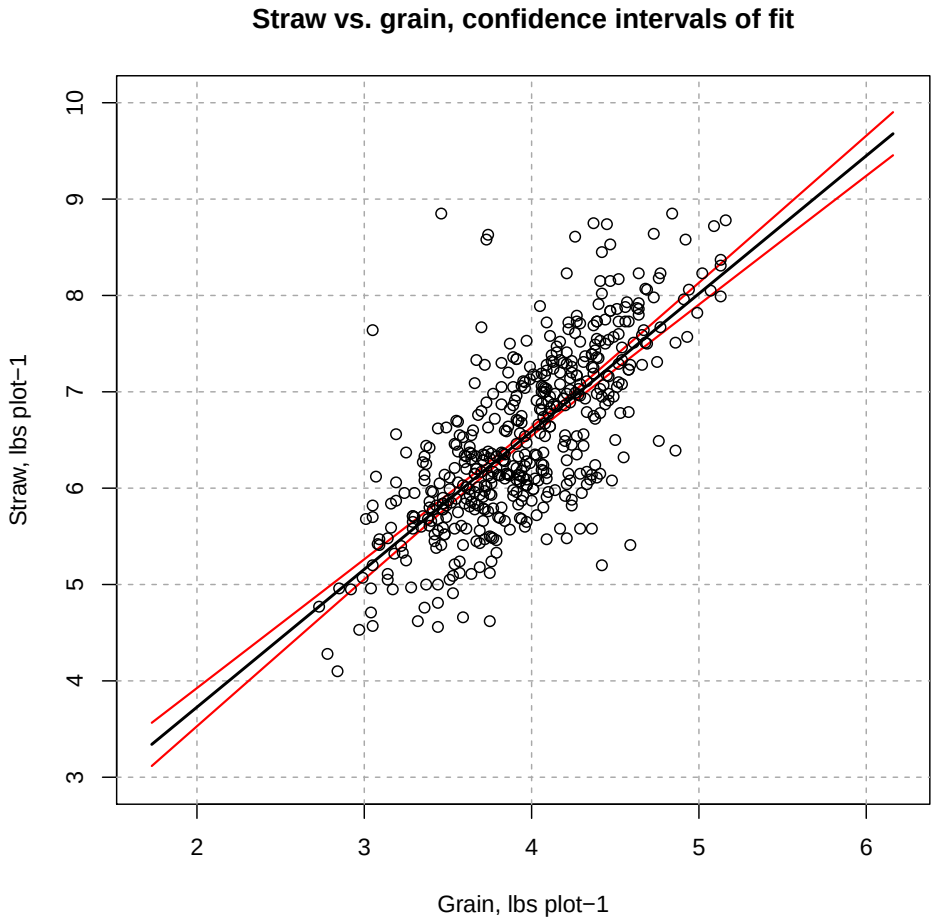
$$s^2_{Y.x} = \frac{1}{n-2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

The variance at other values of the predictand also depends on the **distance from the centroid** $(x_0 - \bar{x})^2$:

$$s^2_{Y_0} = s^2_{Y.x}(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2})$$

This means that the **slope** could "tilt" a bit around the centroid.

Visualization of uncertainty in the regression parameters



In this case, quite a **narrow** confidence range of the **equation**, despite point spread.

Note: R function `predict`, argument `interval="confidence"`

Prediction

One use of the fitted regression equation is to **predict** at arbitrary values of the predictor.

This could apply to **future events** or **observed values of the predictor**, where the **estimated value of the predictand** is wanted.

Example: Grain has been measured but not straw, what is the likely straw yield for a grain yield of 3 lbs plot^{-1}?

Best-fit line: `straw` = 0.87 + 1.43 * `grain`

Direct calculation:

```
> 0.87 + 1.43 * 3

[1] 5.16

[1] "Predicted straw yield for grain yield 3 lbs plot-1: 5.16 lbs plot-1"
```

Prediction uncertainty

Two sources of **prediction uncertainty**:

1. The uncertainty of **fitting** the best regression line from the available data; this is the **estimation** uncertainty (above);

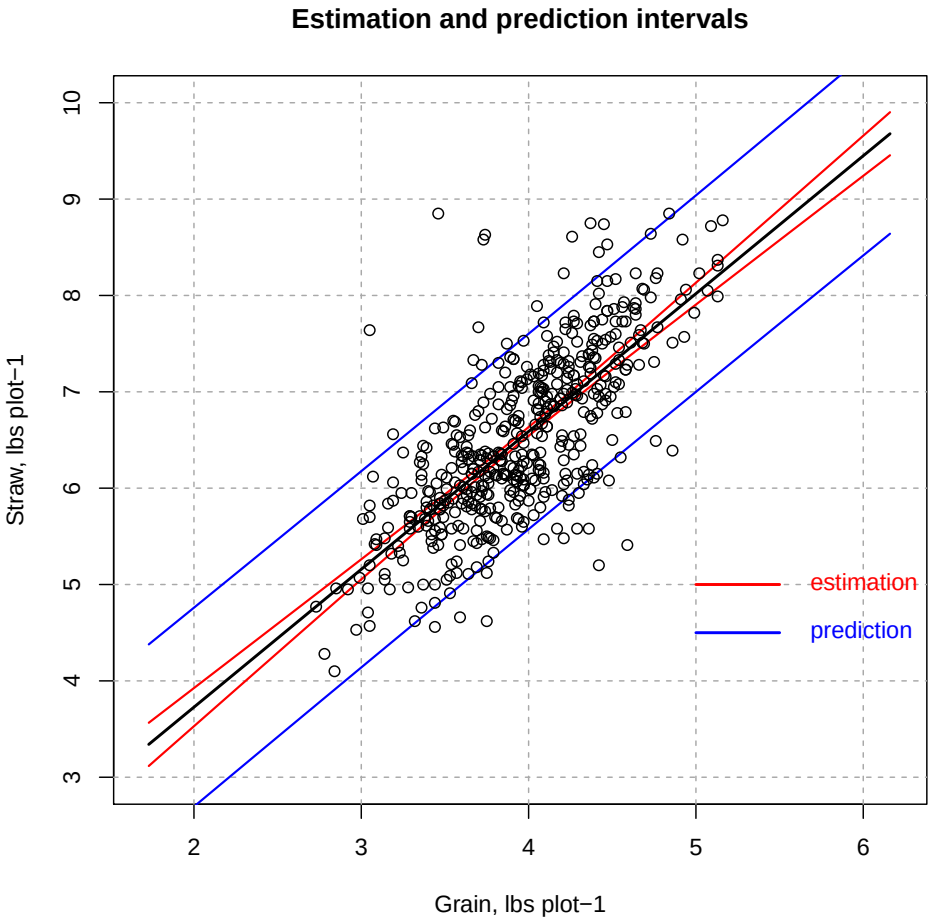
2. The uncertainty in the **process**, i.e. the inherent **noise**: the **residual variance**.

Example: predicted straw yields near centroid (≈ 4), 4.5, 5, 5.5, 6:

```
$fit
     1      2      3      4      5
6.5883 7.3035 8.0188 8.7340 9.4493

$se.fit
       1        2        3        4        5
0.027666 0.043037 0.068863 0.097135 0.126220
```

Notice how the standard error of the fit increases with distance from the centroid.

Visualizing prediction uncertainty



Here, most of the prediction uncertainty is from the **noisy data**, *not* the **fit**.

Note: R function `predict`, argument `interval="prediction"`

Topic: Model evaluation

(Often called "validation")

Measures of model quality:

- **internal** : the data used to build the model is also used to evaluate it
  - goodness-of-fit; adjusted for dataset size and number of parameters, e.g., AIC, adjusted R^2
  - not a true test of predictive accuracy
- **external**: evaluate with **independent** data from the same population
  - a completely different set
  - part of a single set: **split** the dataset into a "calibration" and a "validation" set
- **cross-validation** ("jackknifing")
  - one dataset, repeated split, recalibration, compare predicted with actual

1:1 Evaluation

1. The model is developed using only the observations in the **calibration** set;

2. This model is used to **predict** at the the observations in the **validation** set, using the actual (measured) values of the **predictor** (independent) variable(s);

3. These predicted values are compared to the actual (measured) values of the **response** (dependent) variable in the **validation** set.

This relation should be exactly 1:1

Splitting a dataset

Tradeoff:

1. The calibration set must be large enough reliable modelling;

2. The validation set must be large enough for reliable validation statistics.

A common split in a medium-size dataset (100–500 observations) is 3 to 1, i.e., 3/4 for calibration and 1/4 for validation.

Select observations for each set:

- **random**: select at random (without replacement); this requires no assumptions about the sequence of items in the dataset;

- **systematic**: select in sequence; this requires absence of **serial correlation**, i.e., that observations listed in sequence be **independent**;

- **stratified**: first divide the observations by some factor and then apply either a random or systematic sampling within each stratum, generally proportional to stratum size.

Example: selecting 3/4 for calibration, 1/4 for evaluation

```
> (n <- dim(mhw)[1])

[1] 500

> set.seed(621)
> head(index.calib <- sort(sample(1:n, size = floor(n * 3/4), replace = F)),
+     n = 12)

 [1]  1  2  3  4  6  7  8 10 12 13 14 15

> length(index.calib)

[1] 375

> head(index.valid <- setdiff(1:n, index.calib), n = 12)

 [1]  5  9 11 17 18 21 29 31 34 37 39 41

> length(index.valid)

[1] 125
```

Calibrating the model

The model is built with the calibration subset.

Example: predict straw yield from grain yield, simple linear regression:

```
> cal.straw.grain <- lm(straw ~ grain, data = mhw, subset = index.calib)
> summary(cal.straw.grain)

Call:
lm(formula = straw ~ grain, data = mhw, subset = index.calib)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0145 -0.3451  0.0244  0.3561  3.0500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8258     0.2657    3.11    0.002 **
grain         1.4376     0.0672   21.38   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.607 on 373 degrees of freedom
Multiple R-squared: 0.551,        Adjusted R-squared: 0.55
F-statistic:  457 on 1 and 373 DF,  p-value: <2e-16
```

Predicting at evaluation observations

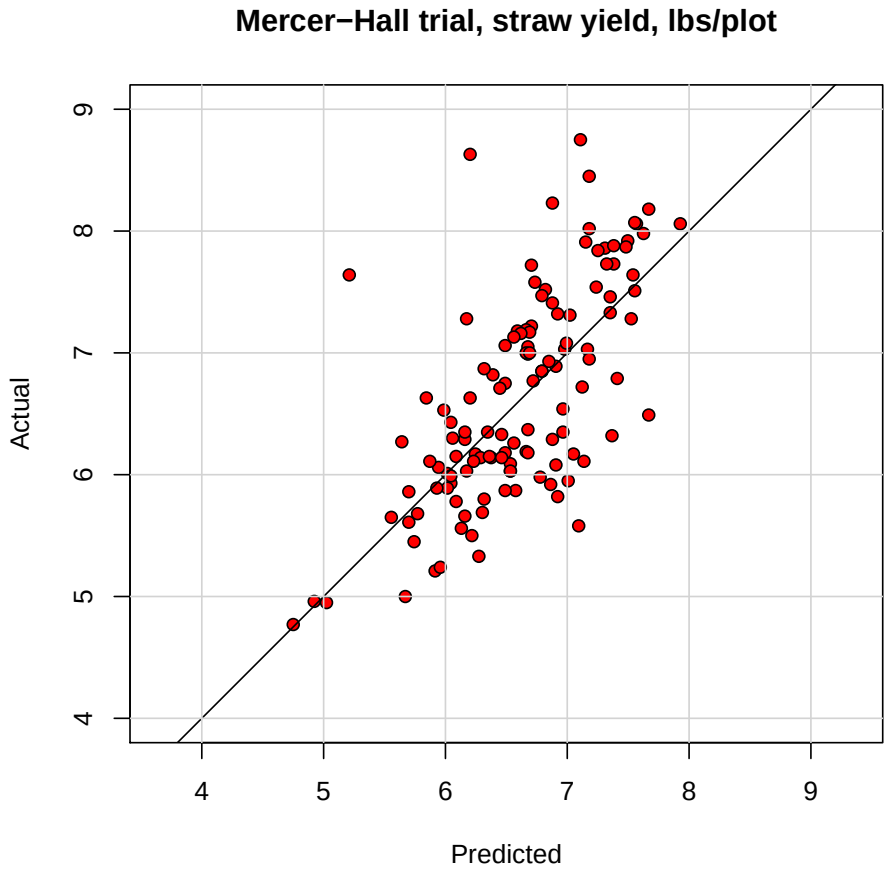
This model is used to predict at the evaluation observations.

```
> summary(pred <- predict.lm(cal.straw.grain, newdata = mhw[index.valid,
+     ]))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.75    6.17    6.66    6.60    7.02    7.93

> summary(actual <- mhw[index.valid, "straw"])

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.77    6.03    6.53    6.65    7.28    8.75
```

Note in this case (typical) the extremes and quartiles are narrower.

Plot on 1:1 line

```
> plot(actual ~ pred, ylab="Actual", xlab="Predicted", asp=1,
+      main="Mercer-Hall trial, straw yield, lbs/plot",
+      xlim=c(4,9), ylim=c(4,9), pch=21, bg="red");
> abline(0,1); grid(lty=1)
```



Note some very poorly-modelled points!

These may reveal model deficiencies (factors not considered).

Measures of model quality

Reference: Gauch, H.G., J.T.G. Hwang, and G.W. Fick. 2003. *Model evaluation by comparison of model-based predictions and measured values*. **Agronomy Journal** 95(6): 1442–1446.

MSD **Mean Squared Deviation**. How close, on average the prediction is to reality.
Square root: Root Mean Squared Error of Prediction (**RMSEP**)

SB **Squared bias**. Are predictions **systematically** higher or lower than reality?

NU **Non-unity slope**. Is the relation between predicted and actual **proportional 1:1** throughout the range of values?
If not, there is either an under-prediction at low values and corresponding over-prediction at high variables (slope > 1), or vice-versa (slope < 1).

LC **Lack of correlation**. How **scattered** are the predictions about the 1:1 line?

$\text{MSD} = \text{SB} + \text{NU} + \text{LC}$

Formulas

n total validation observations; y_i is the true (measured) value of validation observation i; $\hat{y}_i$ is the predicted value of validation observation i; the $\overline{y}$ is the arithmetic mean of the y_i

$$\begin{aligned} \text{MSD} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ \text{SB} &= (\overline{\hat{y}} - \overline{y})^2 \\ \text{NU} &= (1 - b^2)\frac{1}{n}\sum_{i=i}^{n}(\hat{y}_i - \overline{\hat{y}})^2 \\ \text{LC} &= (1 - r^2)\frac{1}{n}\sum_{i=i}^{n}(y_i - \overline{y})^2 \end{aligned}$$

b is the slope of the least-squares regression of actual values on the predicted values, i.e., $\sum y_i\hat{y}_i / \sum \hat{y}_i^2$; this is also called the **gain**.

r^2 is the square of the correlation coefficient $r_{1:1}$ between actual and predicted, i.e., $(\sum y_i\hat{y}_i)^2 / (\sum y_i)^2(\sum \hat{y}_i)^2$.

Geometric interpretation

SB **Translation** The model systematically over- or under-predicts.

- could correct the model with a single consistent translation

NU **Rotation** The average relation between actual and predicted value is not 1:1, after correcting for translation

- typical: rotate below 1:1 – underpredict highest, overpredict lowest values

LC **Scatter** The model is not precise.

These are very different model errors!

Example

```
> paste("SB:", round(valid.sb <- (mean(pred) - mean(actual))^2, 4))

[1] "SB: 0.0024"

> regr.actual.pred <- lm(actual ~ pred)
> paste("NU:", round(valid.nu <- (1 - coef(regr.actual.pred)[2])^2 * mean((pred -
+     mean(pred))^2), 8))

[1] "NU: 0.0005003"

> valid.msd.actual <- mean((actual - mean(actual))^2)
> r2 <- summary(regr.actual.pred)$r.squared
> paste("LC:", round(valid.lc <- (1 - r2) * valid.msd.actual, 4))

[1] "LC: 0.4042"

> paste("MSD:", round(valid.msd <- mean((actual - pred)^2), 4))

[1] "MSD: 0.4071"

> paste("SB + NU + LC:", round(valid.sb + valid.nu + valid.lc, 4))

[1] "SB + NU + LC: 0.4071"
```

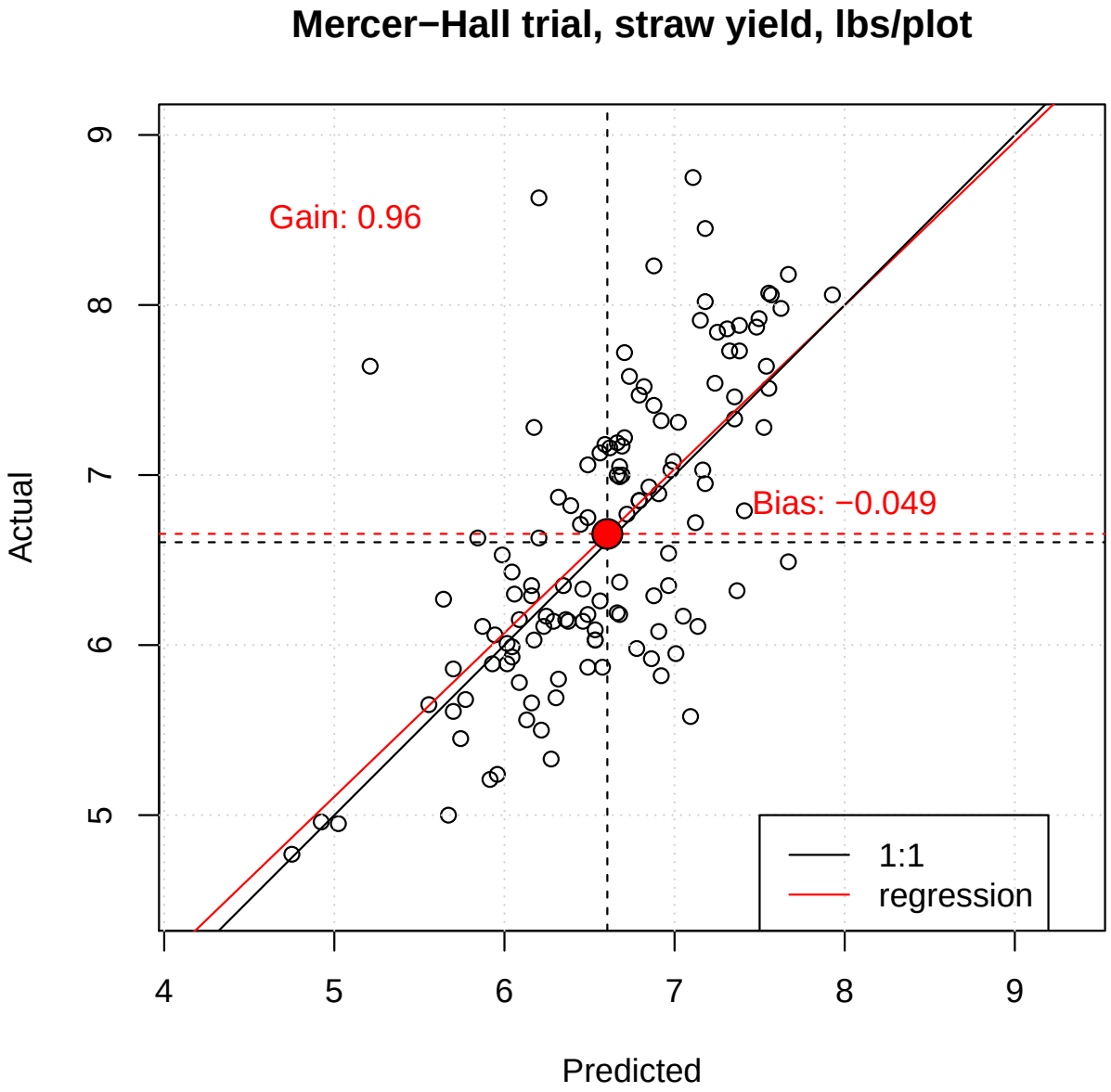
Easily-interpretable measures

```
> paste("Bias:", round((mean(pred) - mean(actual)), 3))

[1] "Bias: -0.049"

> paste("Gain:", round(coefficients(regr.actual.pred)[2], 3))

[1] "Gain: 0.963"

> paste("RMSEP:", round(sqrt(valid.msd), 4))

[1] "RMSEP: 0.6381"
```

Ideally, bias $= 0$, gain $=1$, RMSEP ≈ 0; here:

- slightly negative bias (average under-prediction)
- slightly low gain (typical)
- large RMSEP ($\approx 10\%$ of mean): imprecise model

Visualizing gain and bias



Topic: No-intercept models

It is possible to fit the model without an intercept, i.e., the linear relation is forced through the origin $(0, 0)$. The equation becomes:

$$y_i = \beta x_i + \varepsilon_i$$

There is only a **slope** to be estimated; the intercept is **fixed** at 0.

This is also called **regression through the origin**.

Implications of a no-intercept model

- The mean residual is (in general) not zero;
- The residual sum-of-squares is (in general) larger than for a model with intercept;
- The usual formula for goodness-of-fit is not appropriate (see below).

Even if we know from nature that the relation must incude $(0, 0)$, this takes away a degree of freedom from the fit, and gives a poorer fit.

Appropriateness of a no-intercept model

1. There are **physical reasons** why the relation must include $(0, 0)$;
   - e.g., no straw → no grain is possible (but not vice-versa!)
2. If non-negative variables, a **negative prediction** should be avoided;
   - e.g., impossible to have negative straw or grain in a plot
   - This can also be avoided by setting any negative predictions to zero
3. The **range of the observations** covers $(0, 0)$ or at least is close;
   - otherwise we are assuming a linear form from the origin to the range of our data, when it may have some other form, e.g., exponential, power . . . ; there is no evidence for choosing a linear form **near the origin**
4. The null hypothesis $H_0 : \beta_0 = 0$ in a linear regression *with* intercept can not be disproven (t-test of the coefficient).

Fitting a no-intercept model

The **slope** $\hat{\beta}_{Y.x}$ *can not* be estimated from the sample covariance s_{XY} and variance **of the predictand** s_x^2, because the (co)variances are relative to means, which we can not compute (there is no degree of freedom, because of the fixed intercept).

Instead, the slope is computed by minimizes the RSS, again by **orthogonal projection**: $\mathbf{b} = [\mathbf{x}'\mathbf{x}]^{-1}[\mathbf{x}'\mathbf{y}]$, where the **design matrix** $\mathbf{x}$ here does *not* have an initial column of 1's, just a column of x_i.

This reduces to:

$$\frac{\sum x_i y_i}{\sum x_i^2}$$

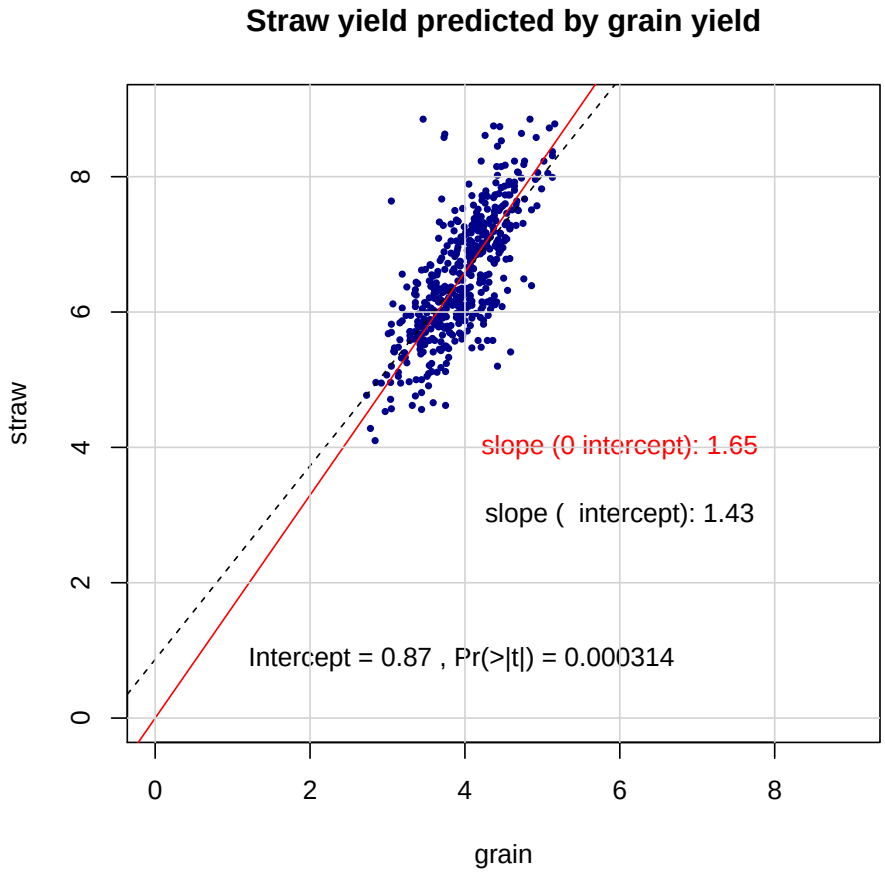
Model summary from no-intercept model

```
Call:
lm(formula = straw ~ grain - 1, data = mhw)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1496 -0.3660  0.0292  0.3657  3.1515

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
grain    1.647      0.007     235   <2e-16

Residual standard error: 0.622 on 499 degrees of freedom
Multiple R-squared: 0.991,        Adjusted R-squared: 0.991
F-statistic: 5.54e+04 on 1 and 499 DF,  p-value: <2e-16
```

The **slope** increased, from 1.43 for the model with intercept to 1.65 for the model without, because the fitted intercept was greater than zero and must be compensated if we force 0 intercept.

The **coefficient of determination** increased substantially, from 0.53 for the model with intercept, to 0.99 for the model without.

Scatterplot with best-fit lines



Here the intercept from the full model is highly unlikely to be zero, so the no-intercept model is not appropriate. Also, the range of the observations is far from $(0, 0)$ so no possibility of negative predictions; no evidence for model form near the origin.

Coefficient of determination for no-intercept model

Since there is no intercept in the design matrix, the **total sum of squares** must be computed relative to zero: $TSS = \sum_{i=1}^{n}(y_i - 0)^2$, rather than relative to the sample mean $\bar{y}$. We still define R^2 as:

$$R^2 \quad = \quad 1 - \frac{\text{RSS}}{\text{TSS}}$$

But since the TSS is computed relative to zero, it tends to be quite high (no compensation for the sample mean), so even though the RSS is larger than if an intercept is included, the R^2 tends to be very high.

Conclusion: R^2 is not a meaningful measure of goodness-of-fit; use residual standard error (or sum-of-squares) instead.

Topic: Structural analysis

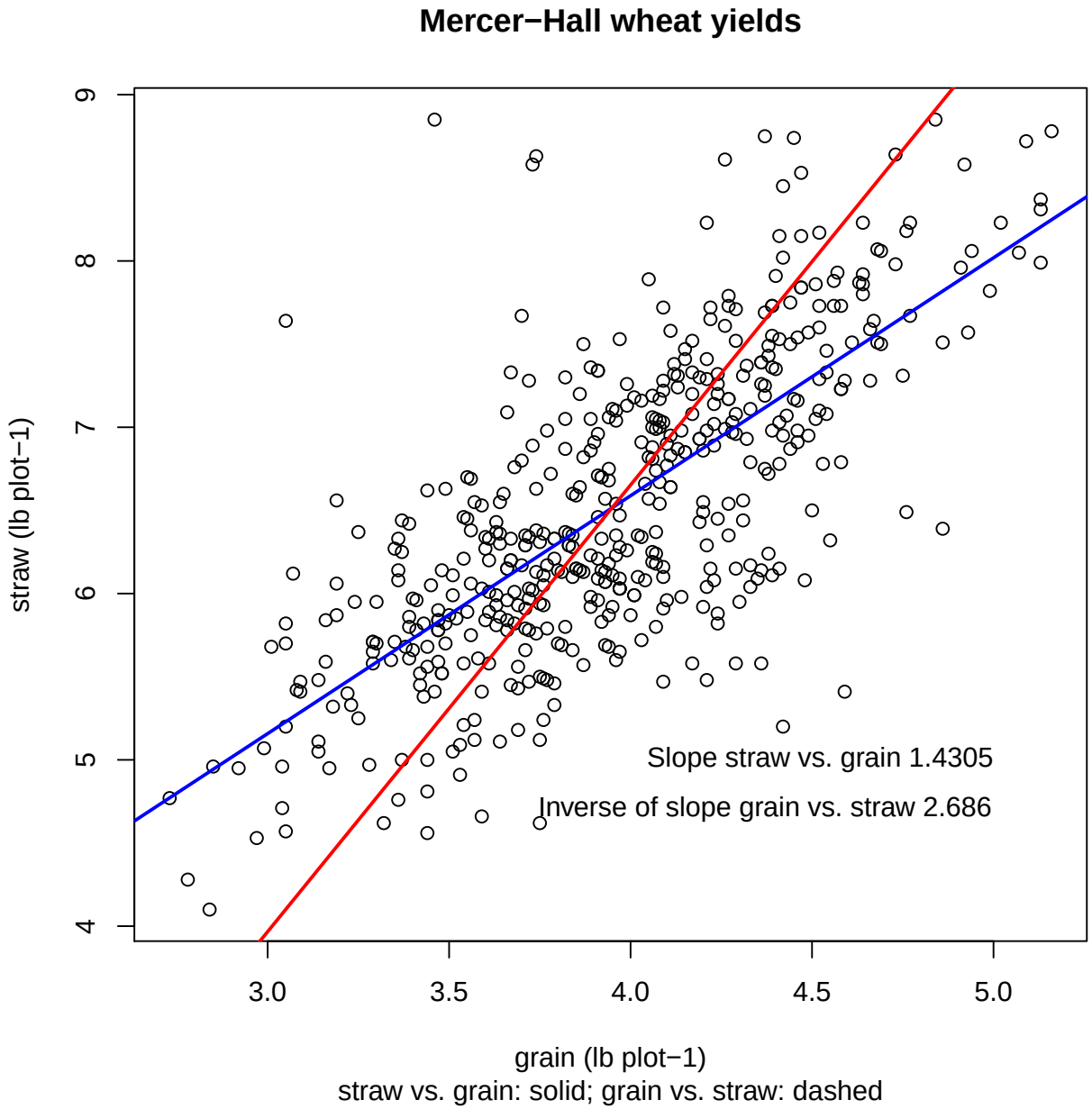
Recall:

1. Variables have **different** status
   (a) A univariate **linear regression** of straw (dependent) on grain (independent) yield;
   (b) A univariate **linear regression** of grain (dependent) on straw (independent) yield.

2. Variables are of **equal** status
   (a) A bivariate **linear correlation** between the two variables (straw and grain yields);
   (b) A **linear structural relation** between the two yields.

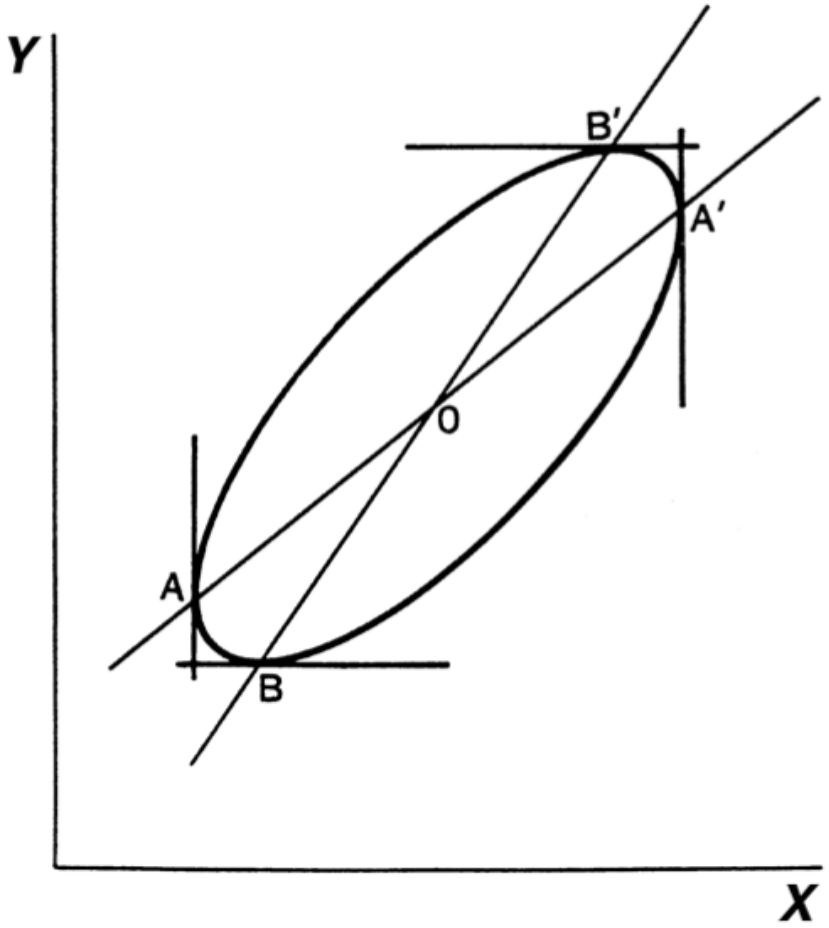
"Structure": underlying relation between two variables, considered equally.

Example: two slopes for the same relation



Random effects lines

Recall:



Source: Webster, *European Journal of Soil Science* **48**:558 (1997), Fig. 1

Which equation is "correct"?

1. If modelling straw based on grain: regression straw vs. grain

2. If modelling grain based on straw: regression grain vs. straw

3. If modelling the **relation between grain and straw**: **structural analysis**

The relation is interesting e.g. for the best description of plant morphology: the grain/straw ratio

Law-like relations

Linear Model (one predictor, one predictand): $y = \alpha + \beta x$

Both random variables have some **random error**, not necessarily the same:

$$X = x + \xi \quad (1)$$

$$Y = y + \eta \quad (2)$$

Error variances σ_ξ^2 and σ_η^2; ratio λ:

$$\lambda = \sigma_\eta^2 / \sigma_\xi^2 \quad (3)$$

Maximum-likelihood estimator of the slope $\hat{\beta}_{Y.X}$ for predictand Y:

$$\hat{\beta}_{Y.X} = \frac{1}{2 s_{XY}} \left\{ (s_Y^2 - \lambda s_X^2) + \sqrt{(s_Y^2 - \lambda s_X^2)^2 + 4 \lambda s_{XY}^2} \right\} \quad (4)$$

Setting the error variance ratio

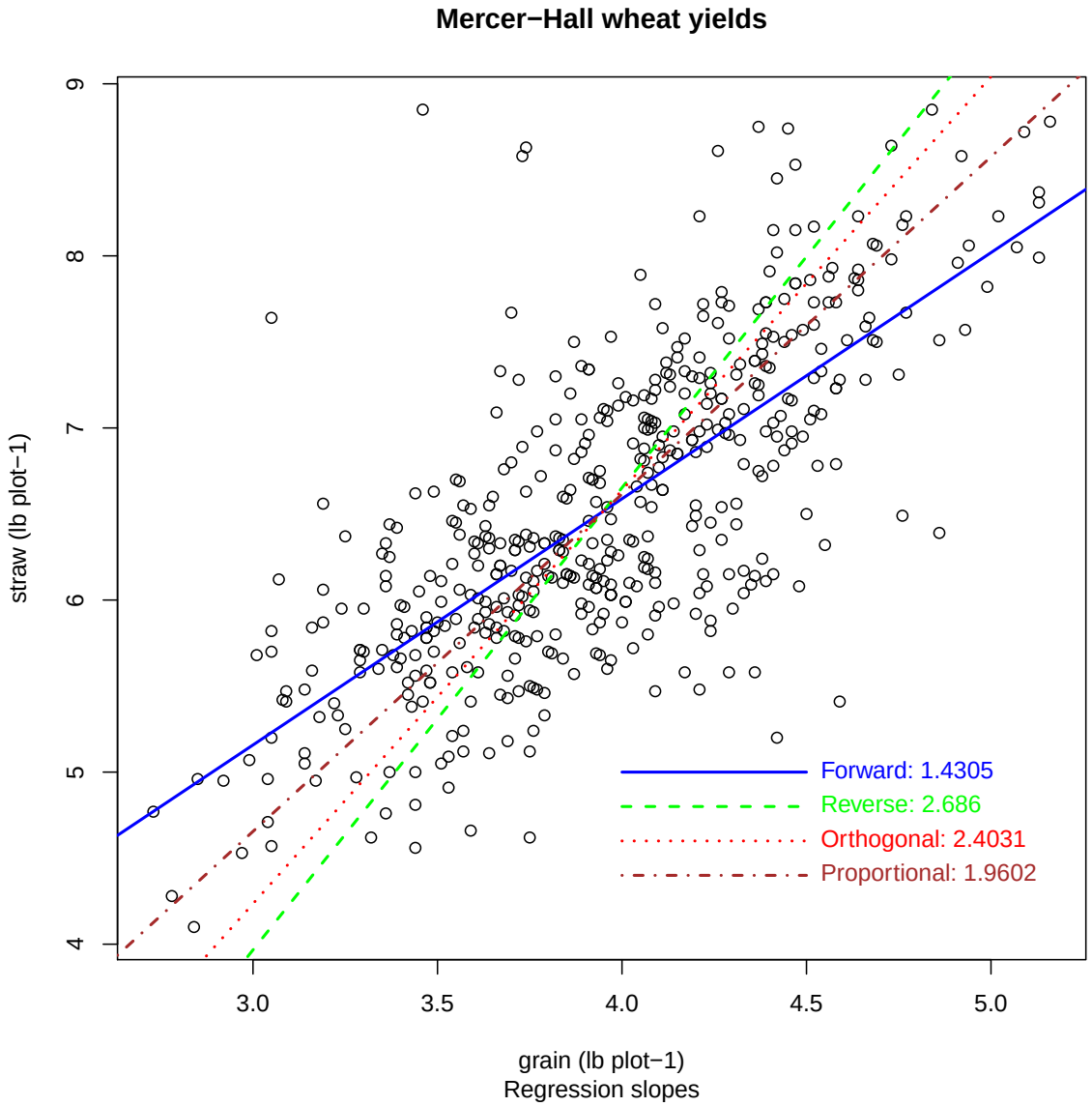
1. From **previous studies**

2. **Orthogonal**: Assume **equal** error variances: $\lambda = 1$
   - must have the same unit of measure
   - must have *a priori* reason to expect them to have similar variability

3. **Proportional**: Equal to the **sample** variances $\lambda \approx s_y^2 / s_z^2$
   - normalizes for different **units of measure** and for different **process intensities**
   - this is the **Reduced Major Axis** (RMA), popular in biometrics
   - It is equivalent to the axis of the first **standardized principal component** (see below)

(In the case of the Mercer-Hall wheat yields, since no treatments were applied by definition $\lambda \approx s_y^2 / s_z^2$ and the RMA should be used.)

Example of structural analysis fits



Topic: Multiple linear regression

Objective: **model** one variable (the **predictand**) from several other variables (the **predictors** or **explanatory** variables)

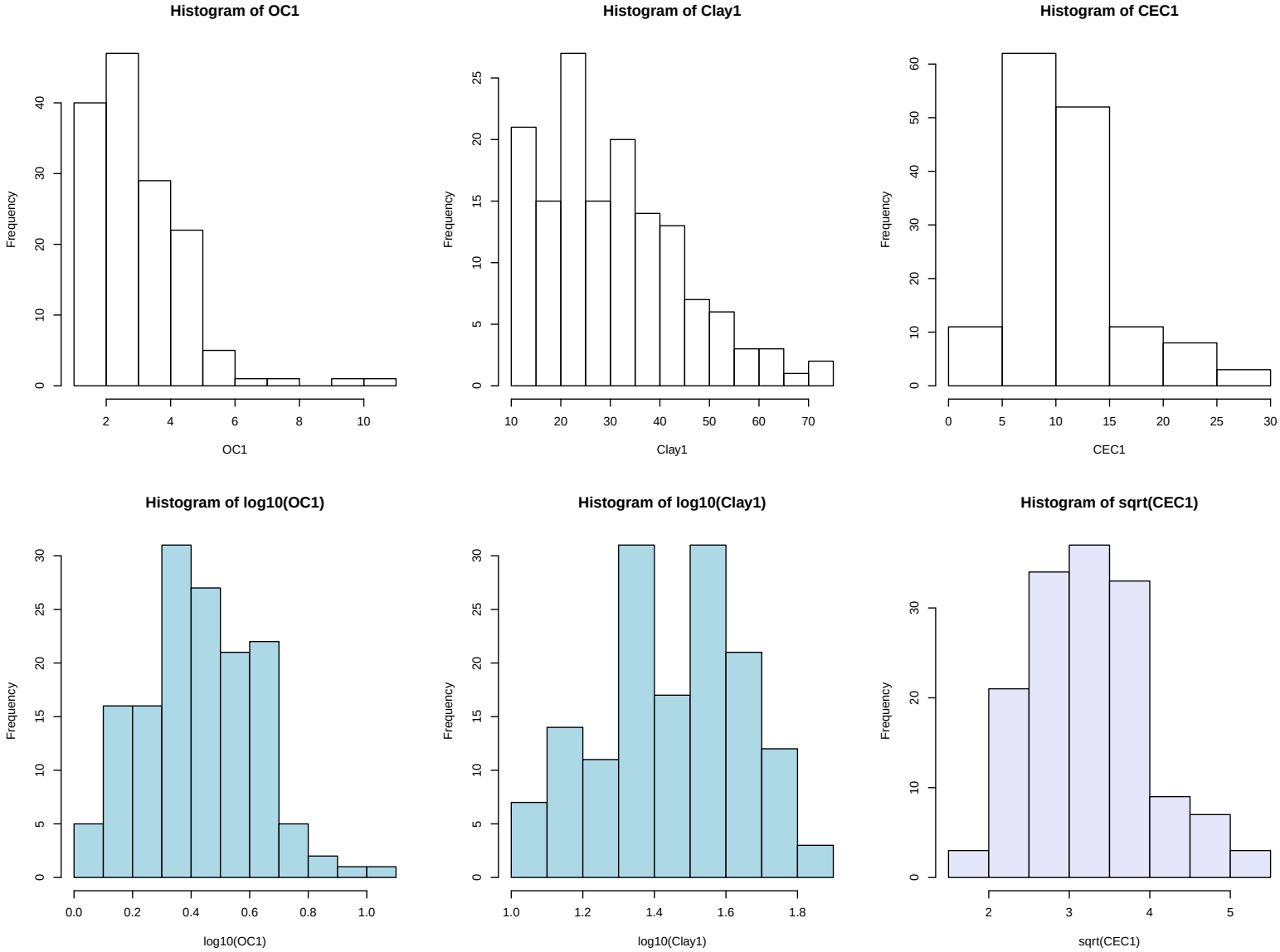
- to **"explain"**
- to **predict**

Example dataset

Source: M Yemefack, DG Rossiter, and R Njomgang. Multi-scale characterization of soil variability within an agricultural landscape mosaic system in southern Cameroon. *Geoderma*, **125**: 117–143, 2005.

- Tropenbos Cameroon research programme
- 147 soil profiles
- geoferenced, in 4 agro-ecological **zones**, 8 **previous landuses**
- Three soil **layers** (1: 0–10 cm, 2: 10–20 cm, 3: 30–50 cm)
- Measured variables:
  1. **Clay content**, weight % of the mineral fine earth (< 2 mm);
  2. **Cation exchange capacity**, $cmol^{+}$ $(kg\ soil)^{-1}$
  3. **Organic carbon** (OC), volume % of the fine earth.

Transform to more symmetric distributions



Example: Modelling CEC

Theory: cations are retained and exchanged by reactive surfaces on clay and organic matter

Objective: explain topsoil CEC by topsoil clay content, topsoil organic matter, **or both**.

Purpose: (1) avoid expensive CEC lab. analysis; (2) understand the process of cation exchange

Models:

1. **null** regression: every value is predicted by the mean.

2. **simple** regressions: CEC = f(clay); CEC = f(OC)

3. **multiple** regression: CEC = f(clay, OC)

   (a) **additive** effects
   (b) **interaction** effects

Model formulas and solution by orthogonal projection

1. $y = \beta_0$
2. $y = \beta_0 + \beta_1 x_1$ (clay)
3. $y = \beta_0 + \beta_1 x_2$ (OC)
4. $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ (clay, OC)
5. $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$ (clay, OC, interaction)

All are solved by **orthogonal projection**:

$$\mathbf{b} = [\mathbf{X}'\mathbf{X}]^{-1}[\mathbf{X}'\mathbf{y}]$$

$\mathbf{b}$: **parameter vector**; $\mathbf{X}$: **design matrix**; $\mathbf{y}$: **response vector**

Correcting for over-fitting

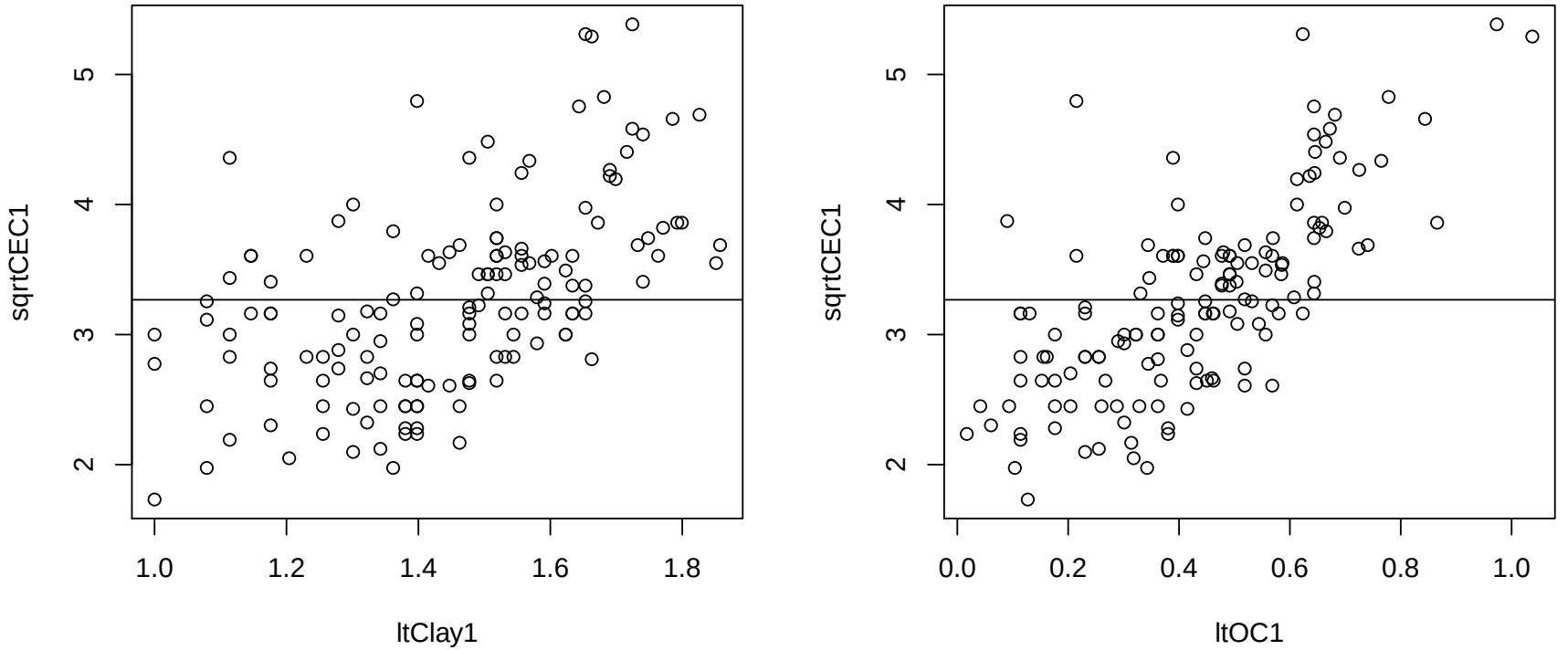
For linear models, use the **adjusted** R^2 in place of the un-adjusted coefficient of determination.

This decreases the apparent R^2, computed from the ANOVA table, to account for the number of predictive factors:

$$R^2_{\text{adj}} \equiv 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

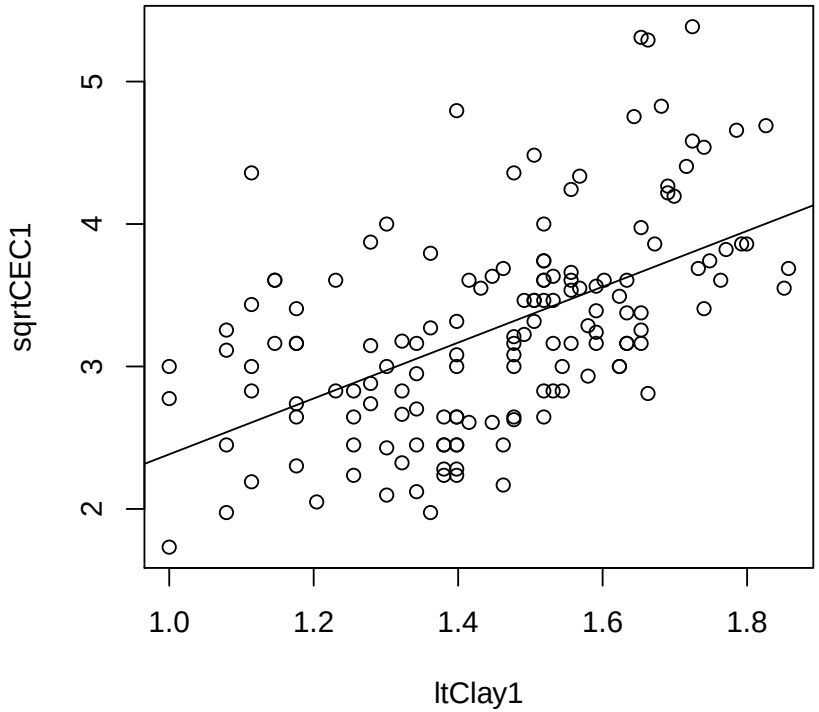
The proportion of variance not explained by the model $(1 - R^2)$ is **increased** with the number of predictors k. As n, the number of observations, increases, the correction decreases.

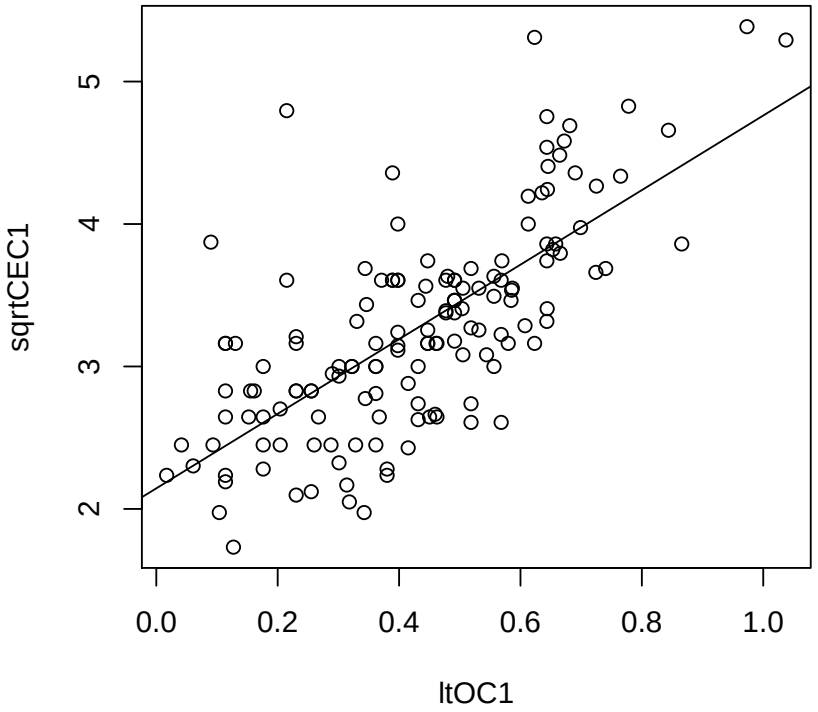
Null model



Adjusted R^2: 0 (by definition: total sum-of-squares is squared deviations from the mean; the mean just **centres** the data)

Simple regression models





Adjusted R^2: 0.2876, 0.5048

Clearly, OC is a much better single predictor than clay

Simple regression models: coefficients

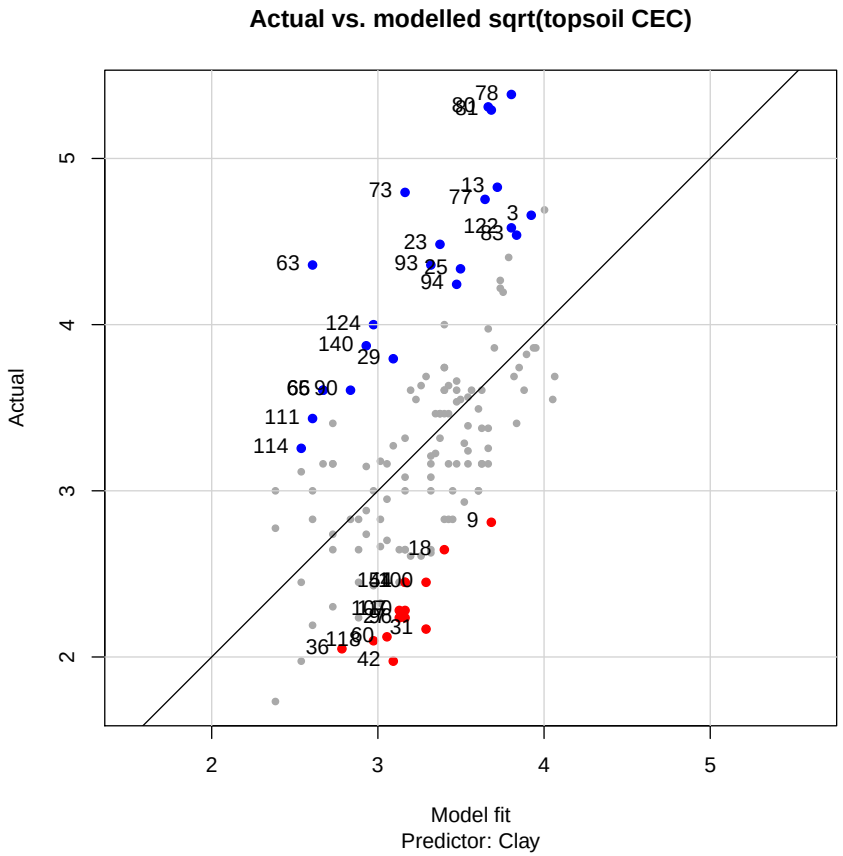
Single predictor: topsoil clay

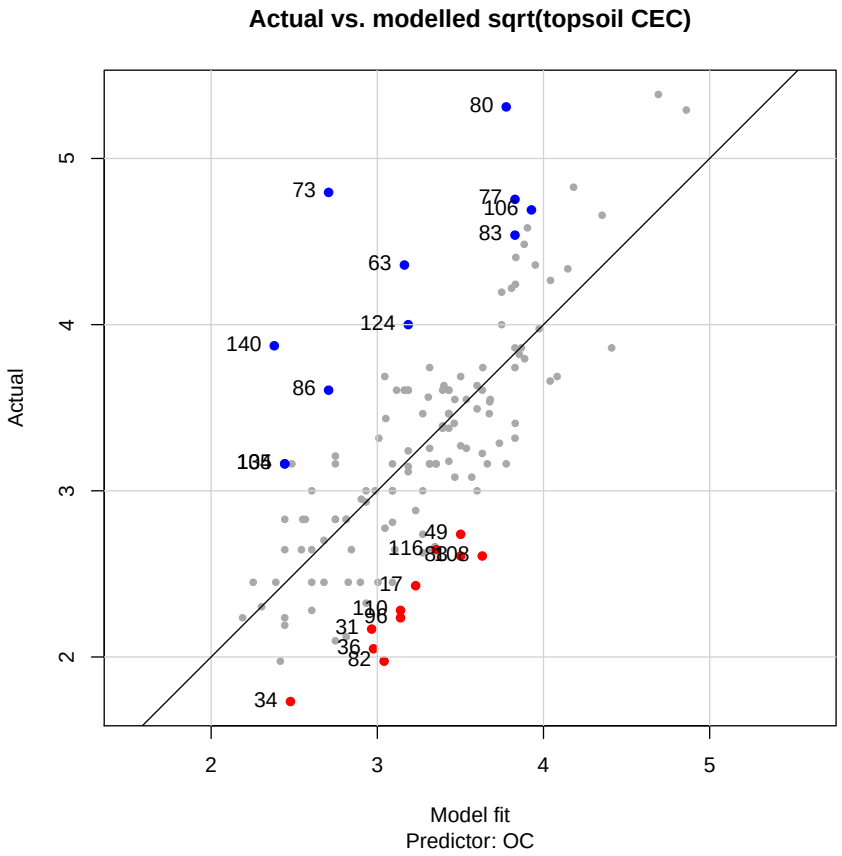
```
Call:
lm(formula = sqrtCEC1 ~ ltClay1)

Coefficients:
(Intercept)      ltClay1
      0.423        1.960
```

Single predictor: topsoil organic C

```
Call:
lm(formula = sqrtCEC1 ~ ltOC1)

Coefficients:
(Intercept)        ltOC1
       2.14         2.62
```

Simple regression models: Actual vs. fits

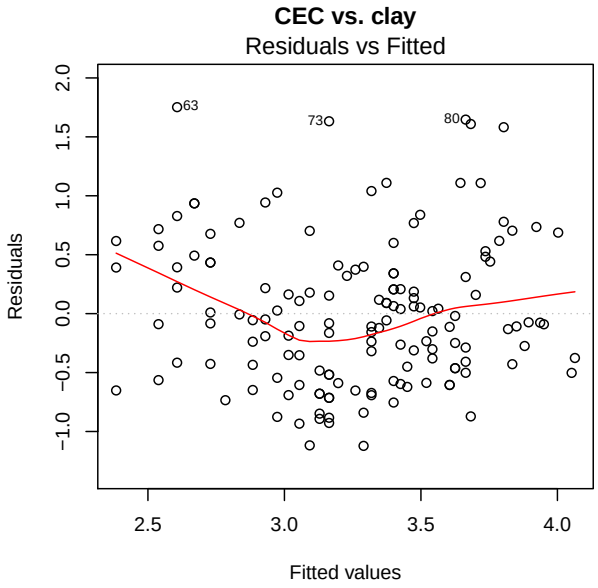


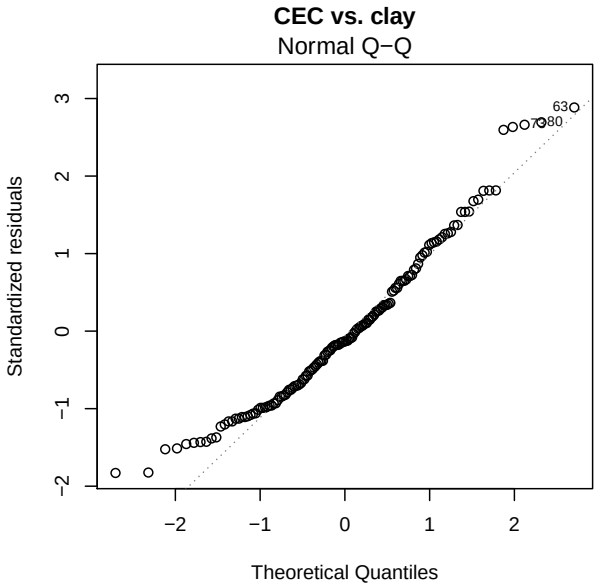


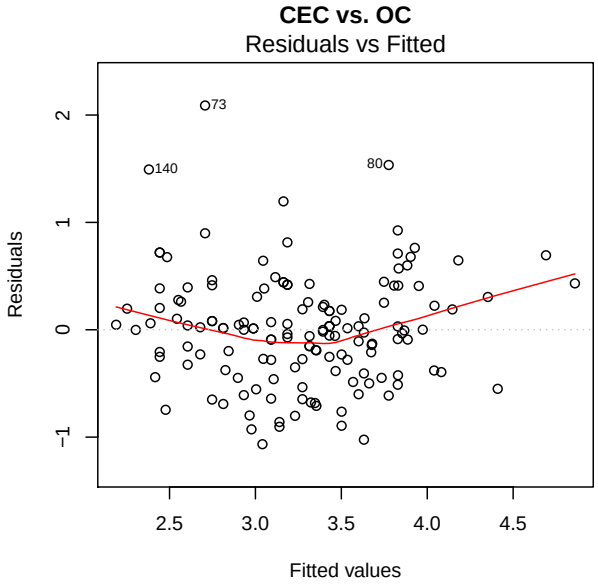
Actual vs. fit are closer to the 1:1 line for the OC predictor model

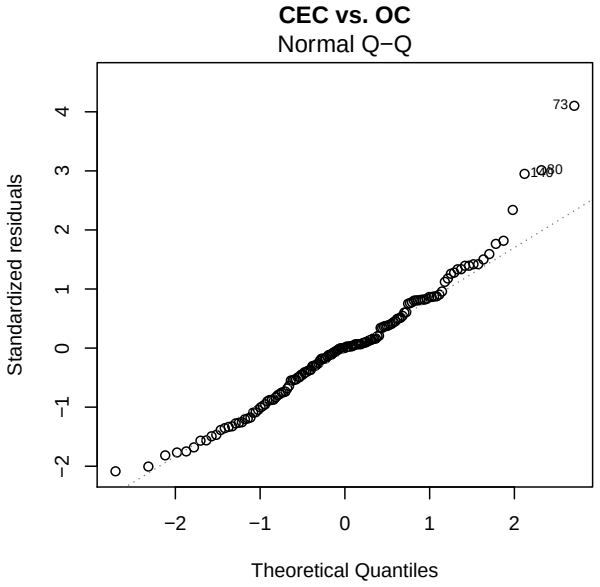
Point cloud is more symmetric around the line

Simple regression models: Regression diagnostics







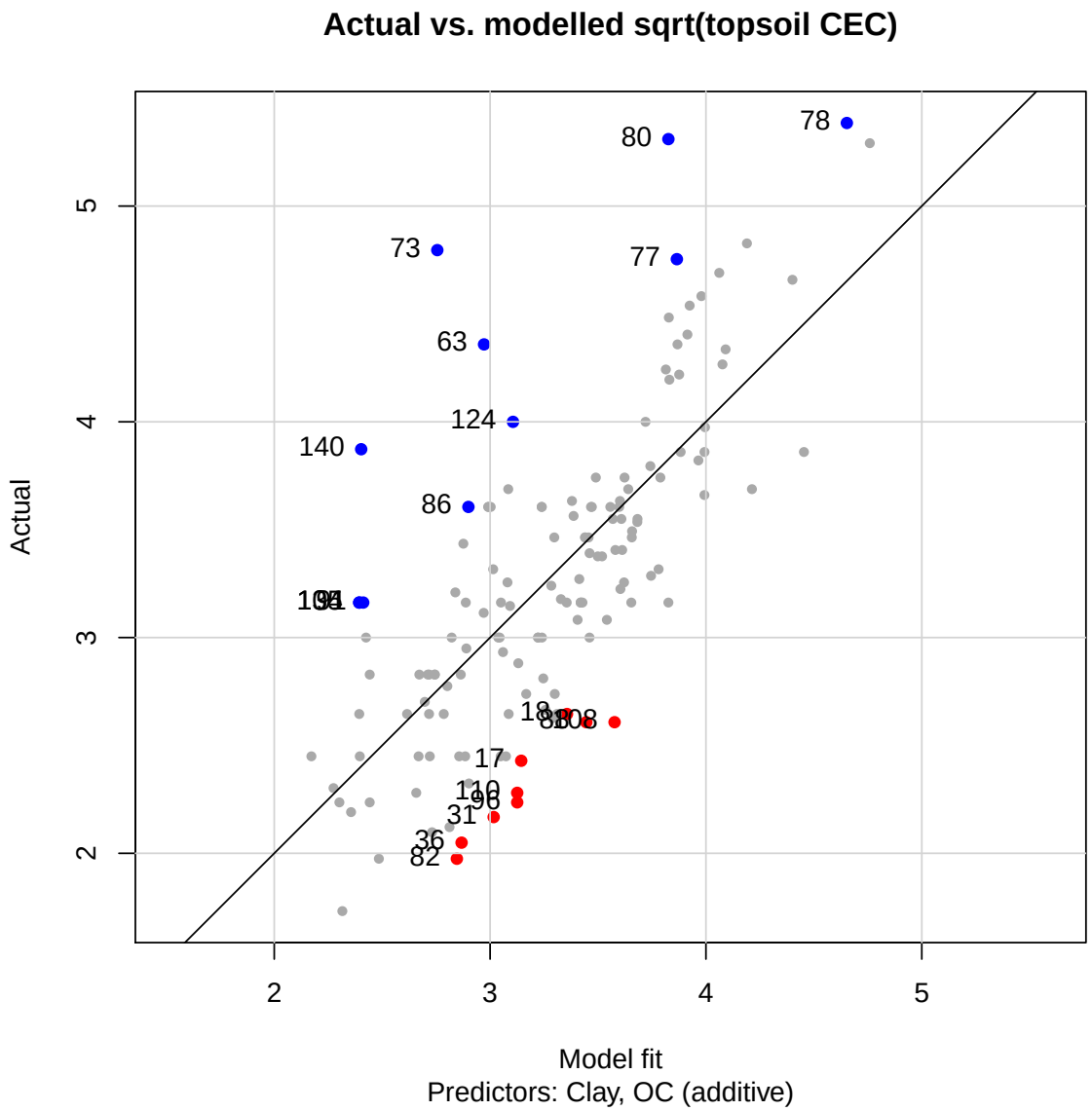


Multiple regression: additive

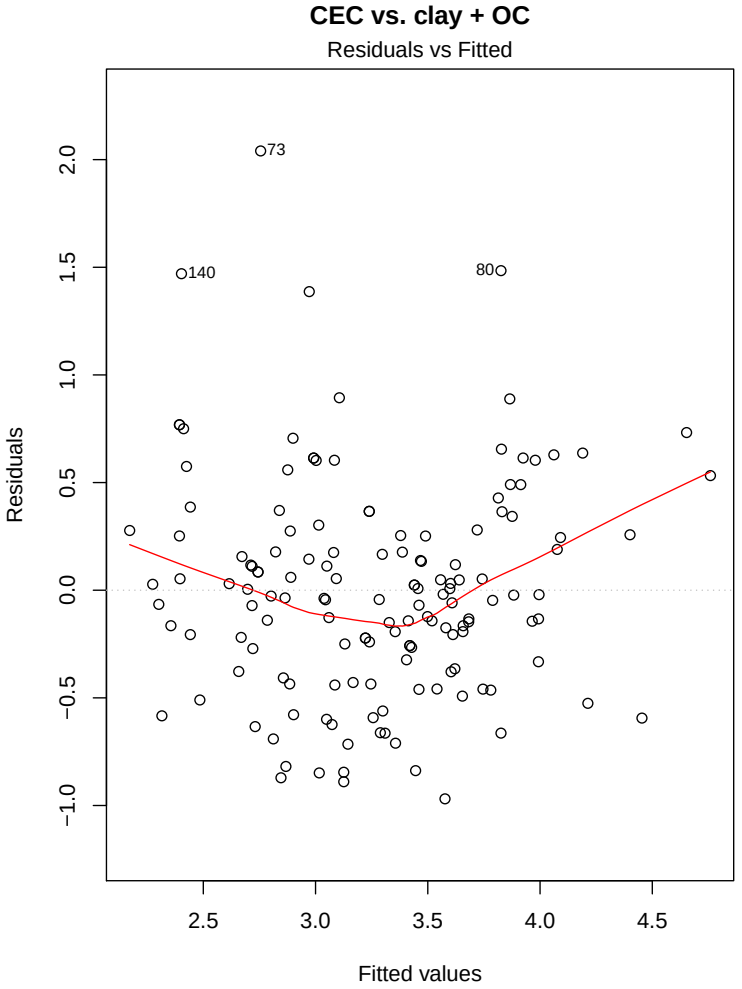
model: CEC = f(clay, OC); Predictors are **independent**

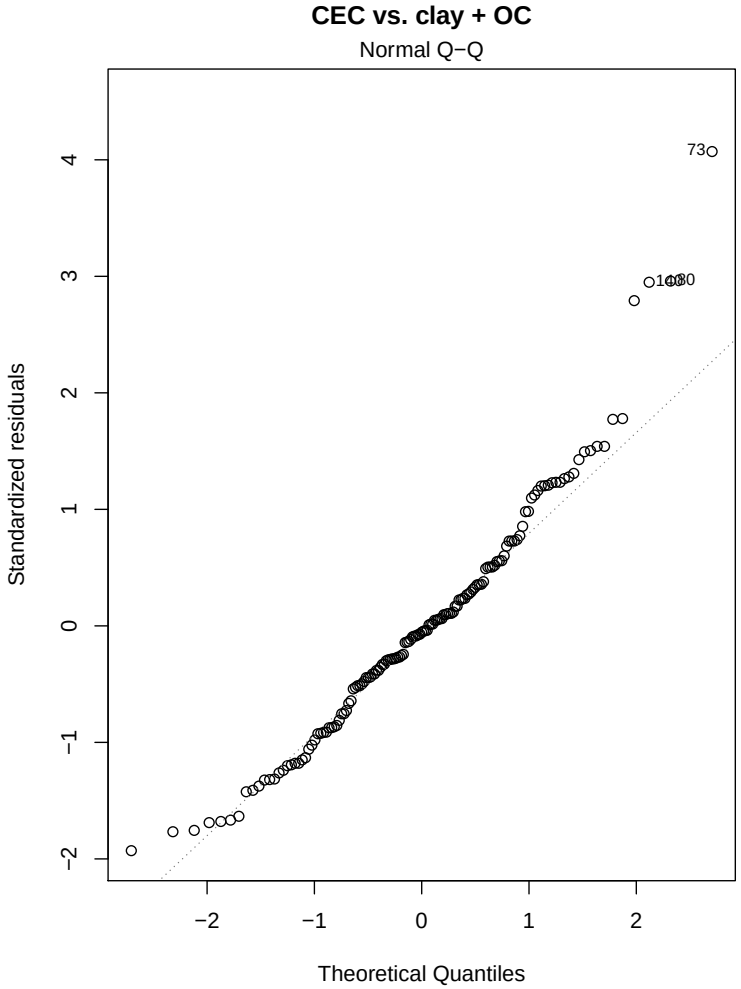
```
Call:
lm(formula = sqrtCEC1 ~ ltOC1 + ltClay1)

Coefficients:
(Intercept)        ltOC1      ltClay1
      1.419        2.239        0.612
```

Additive model: Actual vs. fits



Additive model: regression diagnostics





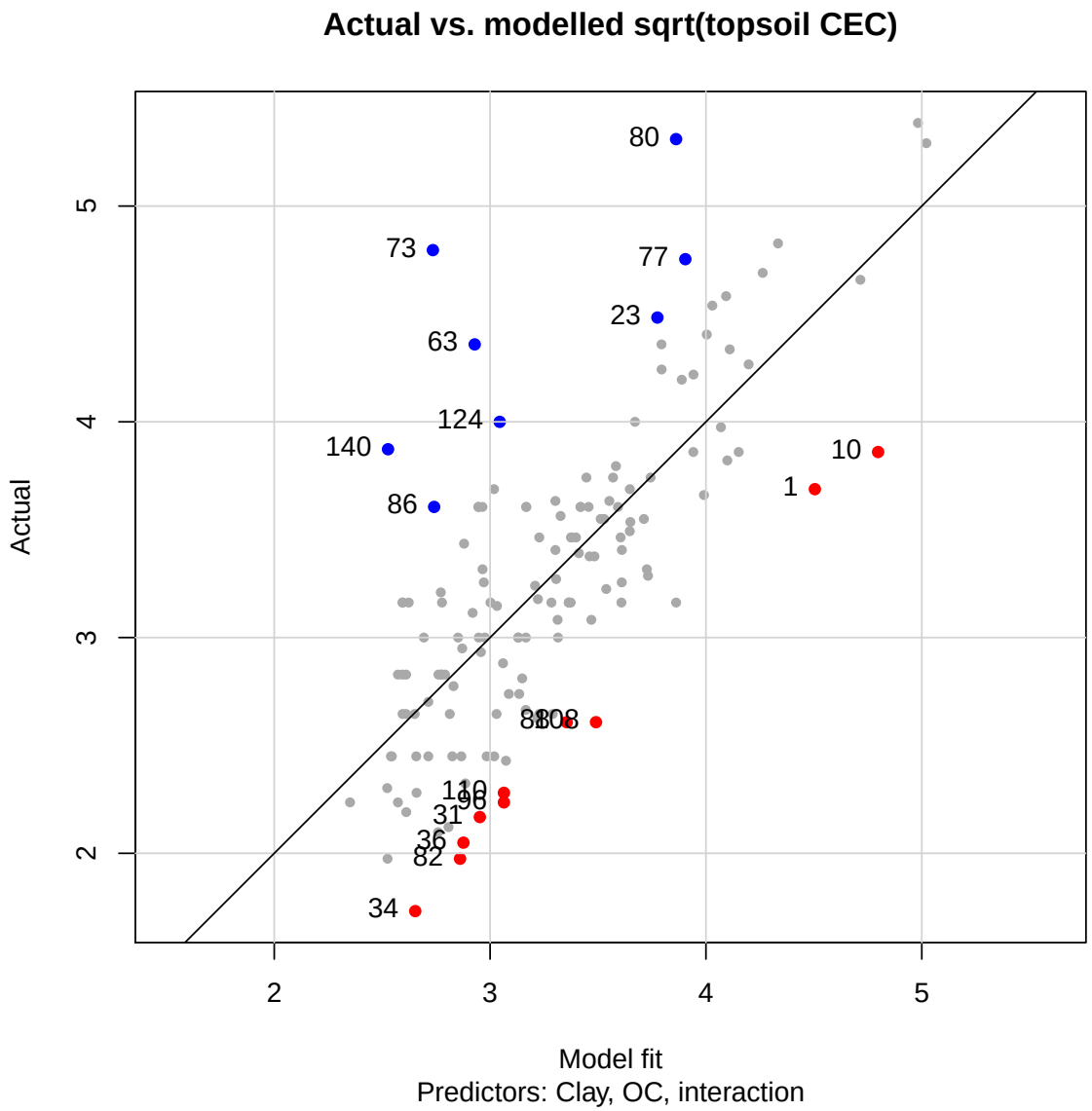
Multiple regression: interaction

model: CEC = f(clay, OC); Predictors may have **interactions**

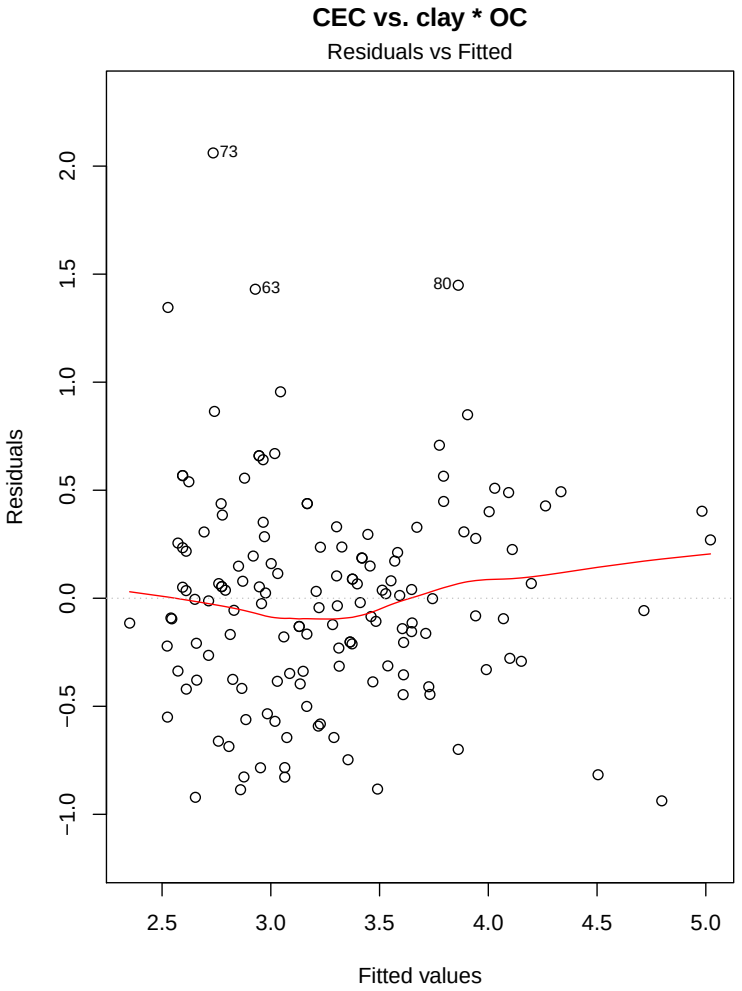
e.g. **synergistic** or **antagonistic** effects

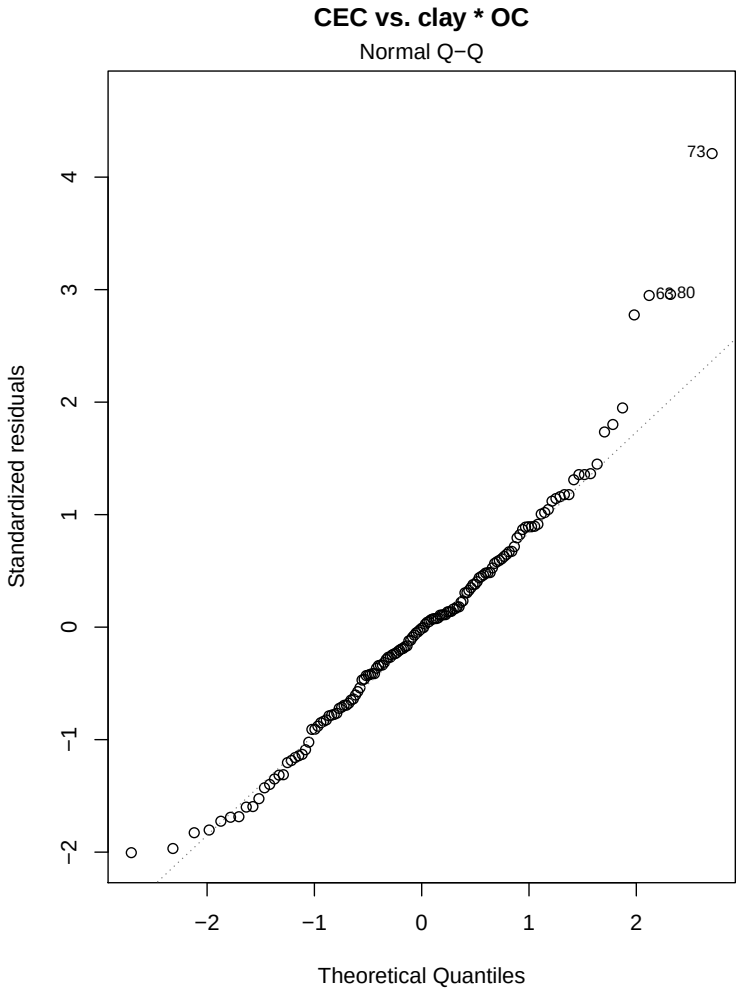
```
Call:
lm(formula = sqrtCEC1 ~ ltOC1 * ltClay1)

Coefficients:
  (Intercept)          ltOC1        ltClay1  ltOC1:ltClay1
        3.158         -2.134         -0.609          2.950
```

Interaction model: Actual vs. fits

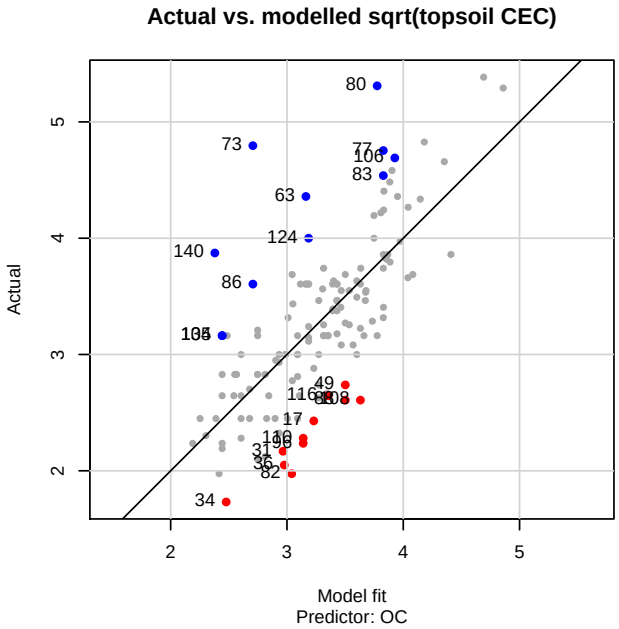


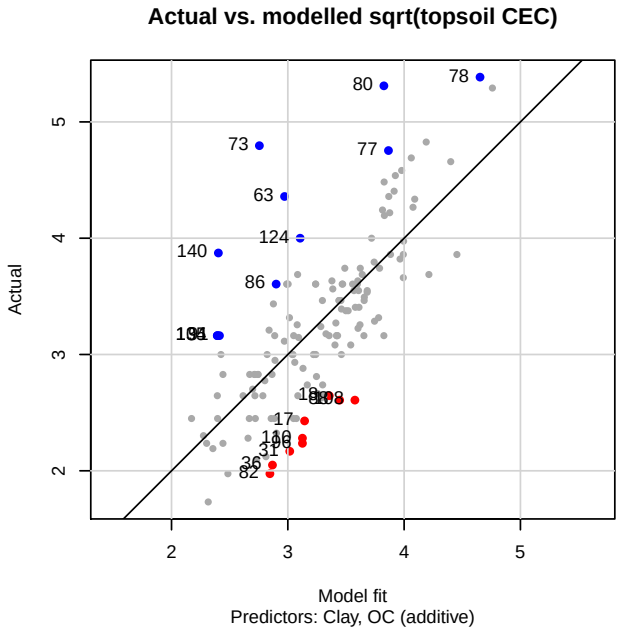
Interaction model: regression diagnostics

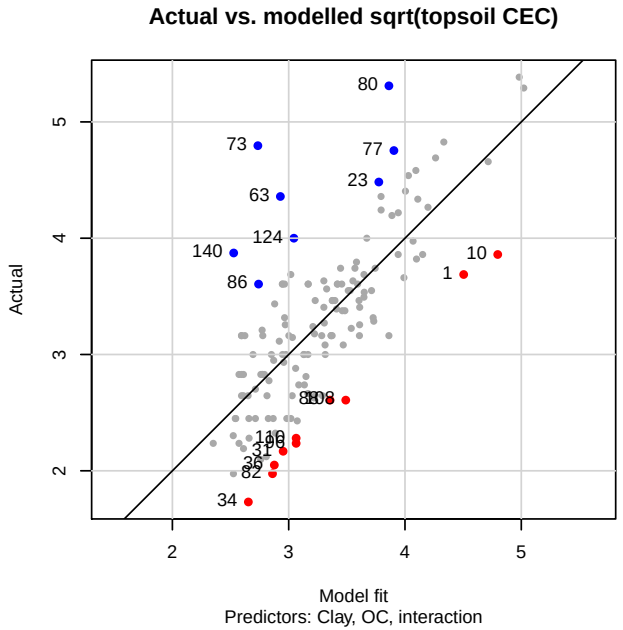




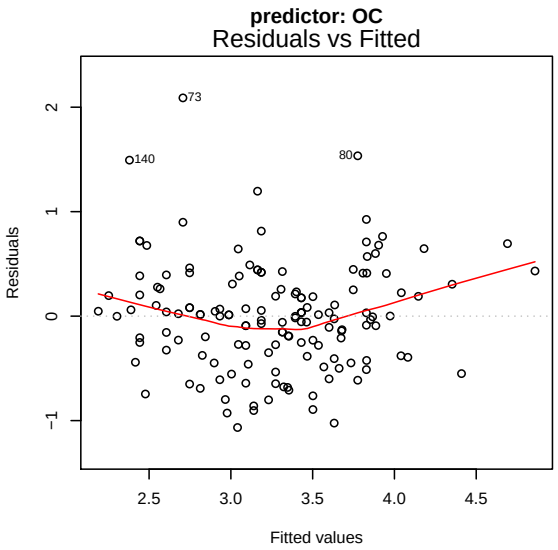
Comparing models – goodness-of-fit

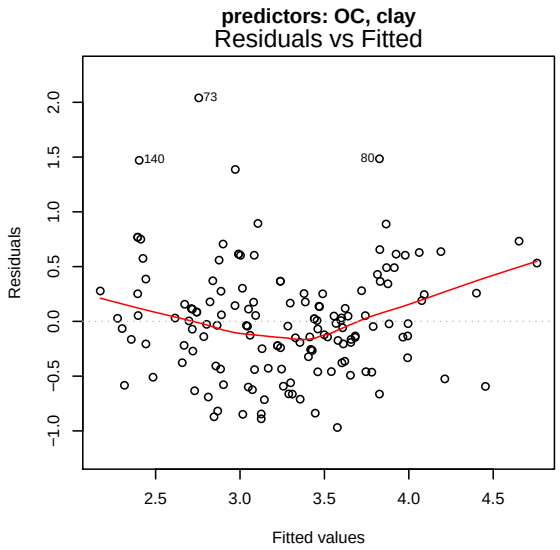


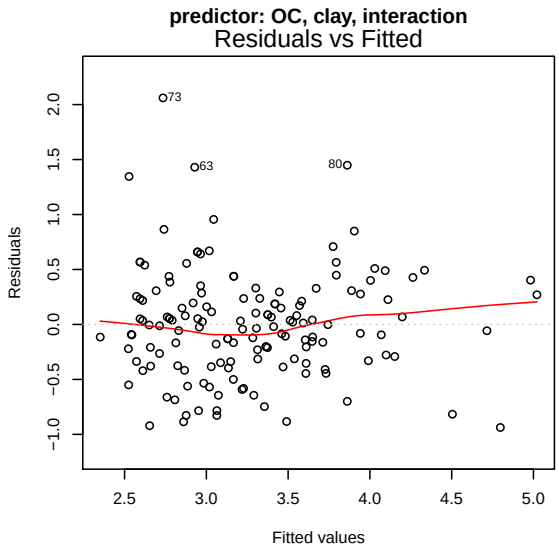


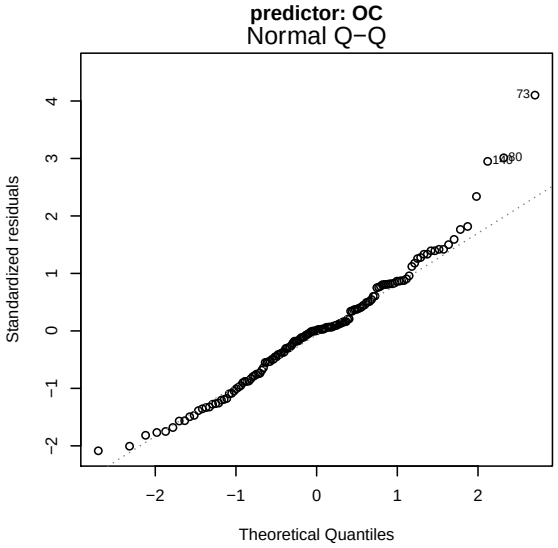


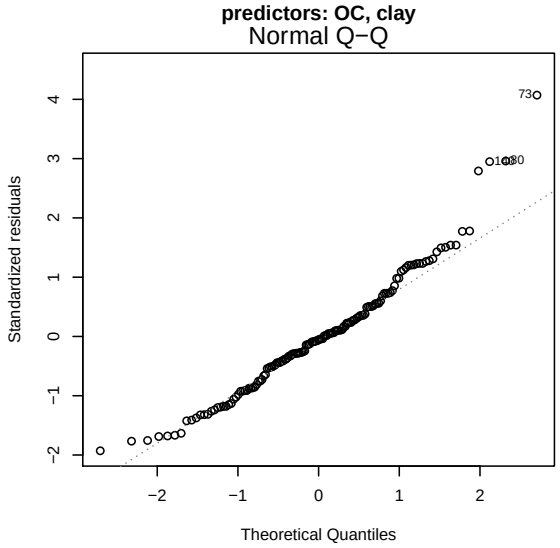
Comparing models – diagnostics

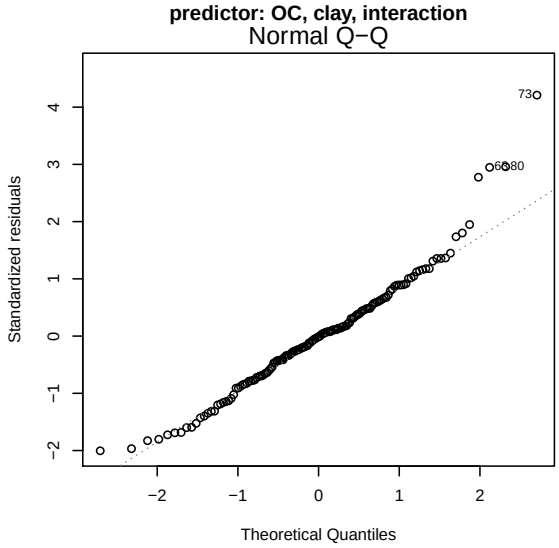












Comparing models – numerically

- Model **summaries**
  - Goodness-of-fit, e.g. adjusted R^2
  - **Significance** of coefficients
- An **Analysis of Variance** of a set of **hierarchical** models
  - Gives the **probability** that the improvement in model (reduction in residual sum-of-squares) is just due to chance

Model summary – simple regression

```
Call:
lm(formula = sqrtCEC1 ~ ltOC1)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0659 -0.3374  0.0012  0.2694  2.0889

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.145      0.101    21.2   <2e-16
ltOC1          2.617      0.214    12.2   <2e-16

Residual standard error: 0.513 on 145 degrees of freedom
Multiple R-squared: 0.508,        Adjusted R-squared: 0.505
F-statistic:  150 on 1 and 145 DF,  p-value: <2e-16
```

Model summary – additive multiple regression

```
Call:
lm(formula = sqrtCEC1 ~ ltOC1 + ltClay1)

Residuals:
   Min     1Q Median     3Q    Max
-0.969 -0.328 -0.027  0.256  2.040

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.419      0.327    4.34  2.6e-05
ltOC1          2.239      0.266    8.43  3.4e-14
ltClay1        0.612      0.262    2.33    0.021

Residual standard error: 0.505 on 144 degrees of freedom
Multiple R-squared: 0.526,        Adjusted R-squared: 0.519
F-statistic: 79.9 on 2 and 144 DF,  p-value: <2e-16
```

Note clay has p=0.0211 probability that removing it from the model (i.e. accepting the null hypothesis of no effect) would be wrong.

In other words, about a 1/50 chance that it doesn't really add to the fit, once OC is in the equation.

Model summary – interaction multiple regression

```
Call:
lm(formula = sqrtCEC1 ~ ltOC1 * ltClay1)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9375 -0.3223 -0.0049  0.2628  2.0610

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.158      0.696    4.54  1.2e-05
ltOC1           -2.134      1.577   -1.35   0.1783
ltClay1         -0.609      0.504   -1.21   0.2295
ltOC1:ltClay1    2.950      1.050    2.81   0.0056

Residual standard error: 0.494 on 143 degrees of freedom
Multiple R-squared: 0.551,        Adjusted R-squared: 0.541
F-statistic: 58.5 on 3 and 143 DF,  p-value: <2e-16
```

Note that the interaction term is here more significant than either single predictor.

ANOVA of a hierarchical set of models

Compare the variance ratios with an **F-test**, taking in account the change in **degrees of freedom**: more for simpler models.

Example: interaction, additive, OC only, null models:

```
Analysis of Variance Table

Model 1: sqrtCEC1 ~ ltOC1 * ltClay1
Model 2: sqrtCEC1 ~ ltOC1 + ltClay1
Model 3: sqrtCEC1 ~ ltOC1
Model 4: sqrtCEC1 ~ 1
  Res.Df  RSS Df Sum of Sq     F Pr(>F)
1    143 34.9
2    144 36.8 -1      -1.9   7.9 0.0056
3    145 38.2 -1      -1.4   5.7 0.0183
4    146 77.6 -1     -39.4 161.8 <2e-16
```

Here the more complex models are all probably better than their hierarchically-simpler models.

Stepwise regression

Automatically decide which predictors to include

- **Forward**: start with best **single-predictor** model, keep **adding** predictors if they "significantly" improve model
- **Backward**: start with **saturated** model (all predictors, all interactions), keep **deleting** predictors if the reduced model is not "significantly" worse

Comparing models: goodness-of-fit, adjusted for number of parameters

Problem: if there is (near)**colinearity** selection of predictors can be sensitive to just a few data points

Problem: can substitute for **modeller's judgement**, especially if several models give similar results

Example of stepwise regression

Predict **CEC in the 30-50 cm layer** . . .

. . . from all three variables (clay, OC, and CEC) for the two **shallower** layers

i.e. total of **six** possible predictors – are all necessary?

(Purpose: avoid sampling the deeper subsoil)

Final results are different!

Forward:

```
Call:
lm(formula = Clay5 ~ ltClay1 + Clay2 + CEC2)

Coefficients:
(Intercept)      ltClay1        Clay2         CEC2
      9.402        5.313        0.798       -0.235

[1] "AIC:  835.9"
```

Backward:

```
Call:
lm(formula = Clay5 ~ Clay2 + CEC2)

Coefficients:
(Intercept)        Clay2         CEC2
     14.519        0.861       -0.199

[1] "AIC:  835.2"
```

Topic: Regression trees

Objective: **model** one variable (the **predictand**) from several other variables (the **predictors** or **explanatory** variables)

This is the same objective as for MLR and other model-based regression methods, **but**:

- no need to choose the functional form (e.g., multivariate linear)
- no assumption that the functional form is the same throughout the range of the predictors.
- no need to transform predictors or predictand to satisfy the assumptions of a model form
- no need to choose among correlated predictor variables
- no need to explicitly consider (or not) interactions

Data mining vs. statistical modelling

This is a **data mining** approach: do not impose a statistical model, rather, propose an **algorithm** to reveal the **structure** in the dataset.

Here the structure is a **binary tree** such that each split improves the prediction:

- by the maximum **reduction** in **within-group** variance
- this is equivalent to the maximum **increase** in **between-group** variance.

The **leaves** (terminal nodes) each then have a **simple prediction model**, usually a **constant** that is the predicted value for all cases that end at that terminal node..

The tree can easily be **interpreted**: we see the variables and their threshold values, and can follow the tree for any new observation. .

Regression trees algorithm

1. Identify the predictors and predictand; compute the overall mean and variance of the predictand.

2. Recursively:

   (a) Look for the **predictor variable**, and its **threshold value**, that "best" splits the data into **two** groups.
   - "Best": maximum reduction in sum of within-group sums of squares in the response variable: $SS_T - (SS_L + SS_R)$.

   (b) Split at that point into two **subtrees**

   (c) Compute the mean and variance of the predictand in each group

3. This continues until the subgroups either:

   (a) reach a user-specified **minimum size**, or

   (b) **no substantial improvement** can be made; that is the sum of the within-groups sum of squares can not be further reduced below a user-defined threshold.

Example: A regression tree for Cameroon CEC

Recall: predict cation exchange capacity (CEC) of topsoils from their organic C and clay concentration.

Fit a **full tree** using the two predictors. Note there is (and can not be) any interaction term.

```
> library(rpart)
> tree <- rpart(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs, xval=20, minsplit=4, cp=0.0075)
> x <- tree$variable.importance; (variableImportance = 100 * x / sum(x))

  ltOC1 ltClay1
 69.738  30.262
```

The last line shows the relative **importance** of each variable in making the prediction, i.e., how much variance was reduced by the splits based on each variable. Here we see OC is twice as important as clay in predicting CEC in this sample set.

Control parameters

Arguments to `rpart.control`, passed from `rpart`:

`minsplit` minimum number of observations at a leaf to try to split

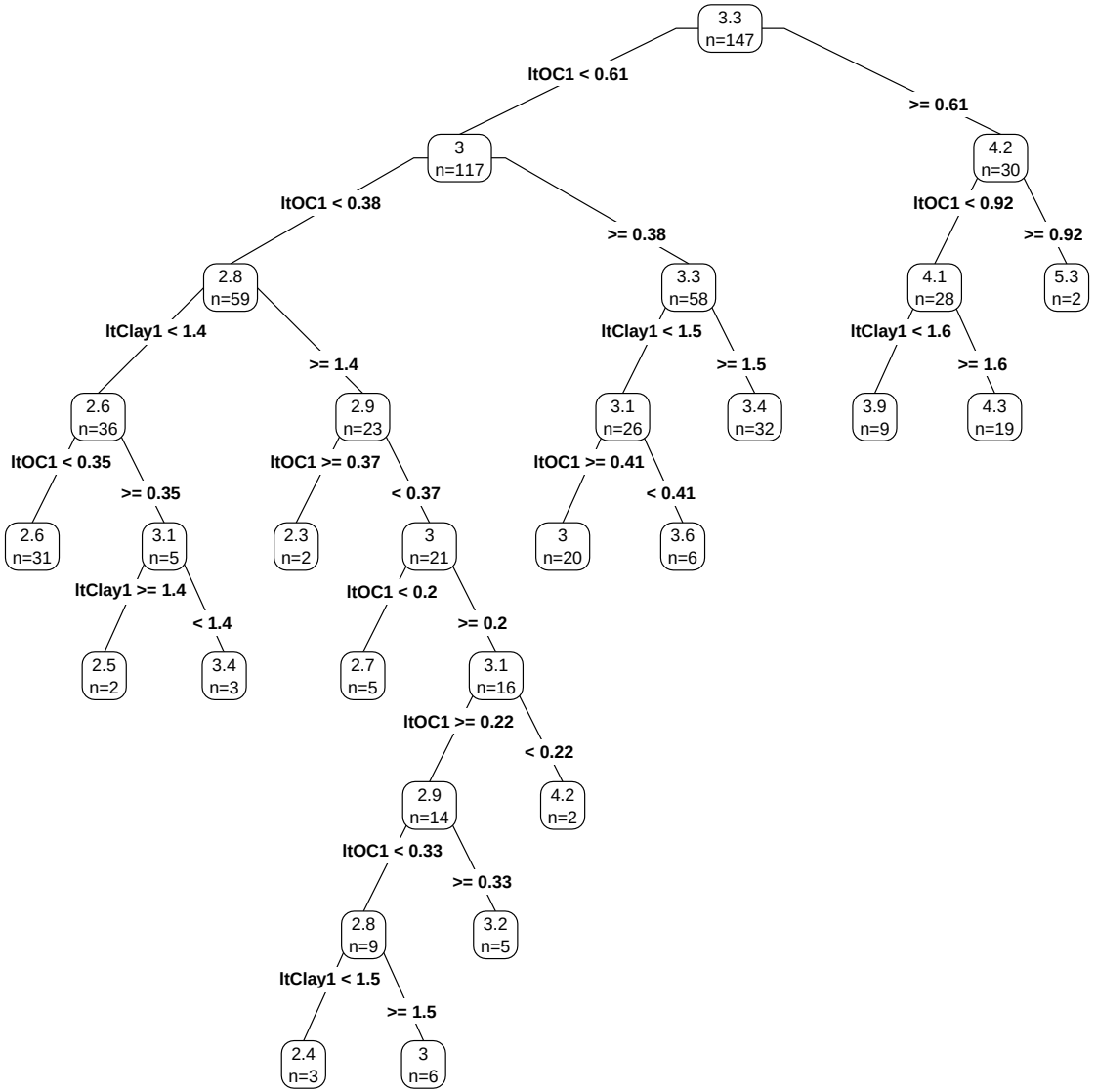
`cp` complexity parameter, see "pruning", below

`xval` number of groups for cross-validation, see "pruning", below

The next slide shows the full tree.

```
> library(rpart.plot)
> rpart.plot(tree, type=4, extra=1)
```

Full regression tree



- **Leaves**: number n of observations; mean value of the predictand at these
- **Branches**: selection variable and threshold value
- **Root**: all observations and their mean value ("null model")

Assessing over-fitting

A full tree over-fits: it fits **noise** specific to this dataset, i.e., this **sample**, rather than **structure**, common to all datasets that could be collected from the underlying **population**.

Assess this with x-fold **cross-validation**, to find the optimum tree size, we then **prune** the tree to this size. Algorithm:

1. Randomly split the observations into x groups (`rpart.control` default is 10)..

2. For each complexity parameter (roughly, the maximum number of splits):

   (a) For each group:
      i. Remove from the dataset
      ii. Re-fit the tree without the removed observations
      iii. Use the tree to predict at the removed observations, using their predictor values
      iv. Compute the squared error
   (b) Summarize errors as root-mean-squared error (RMSE).

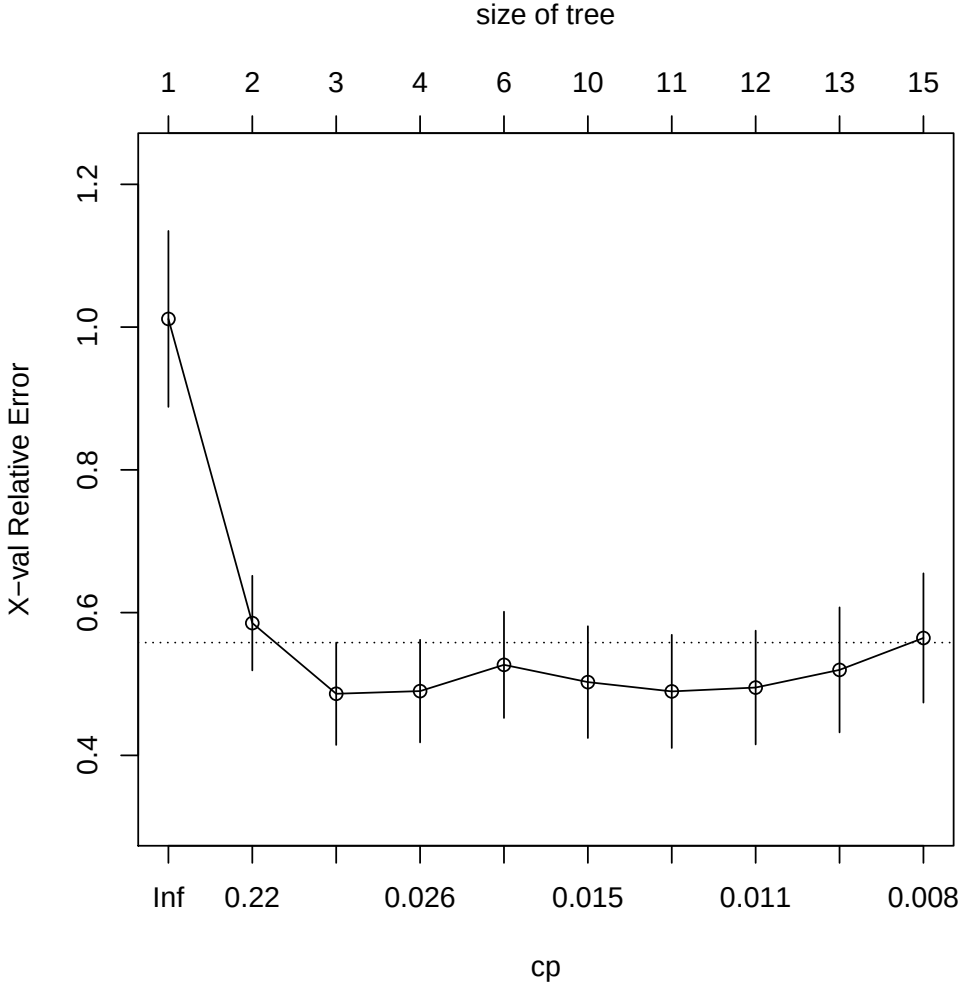
3. Display a table and graph of complexity parameter vs. cross-validation error

Control parameter vs. cross-validation error: table

```
> printcp(tree) # this will be slightly different with each call to rpart: random split for x-val

Regression tree:
rpart(formula = sqrtCEC1 ~ ltOC1 + ltClay1, data = obs, xval = 20,
    minsplit = 4, cp = 0.0075)

Variables actually used in tree construction:
[1] ltClay1 ltOC1

Root node error: 77.6/147 = 0.528

n= 147

        CP nsplit rel error xerror   xstd
1  0.44346      0     1.000  1.011 0.1233
2  0.11258      1     0.557  0.585 0.0663
3  0.03435      2     0.444  0.486 0.0717
4  0.02035      3     0.410  0.490 0.0719
5  0.01808      5     0.369  0.527 0.0744
6  0.01323      9     0.297  0.503 0.0783
7  0.01126     10     0.283  0.490 0.0793
8  0.01102     11     0.272  0.495 0.0797
9  0.00845     12     0.261  0.520 0.0876
10 0.00750     14     0.244  0.564 0.0905
```

Control parameter vs. cross-validation error: graph

```
> plotcp(tree) # this will be slightly different with each call to rpart: random split for x-val
```



Here it seems we only need a 3-split tree!

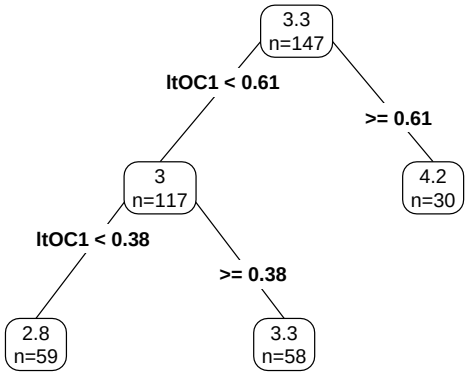
The data was very noisy with respect to these two predictors.

Examine the previous table or this graph to find the complexity parameter corresponding to this number of splits.

Correcting for over-fitting

Prune the tree back to the value of the complexity parameter suggested by the cross-validation plot:

```
> ix <- which.min(tree$cptable[,"xerror"])  # find the minimum cross-validation error
> ix.cp <- tree$cptable[ix,"CP"] # associated complexity parameter
> tree.p <- prune(tree, cp=ix.cp)  # prune to this complexity

> rpart.plot(tree.p, type=4, extra=1)
```



Only OC is now used; there are only three groups of CEC

Prediction with a regression tree

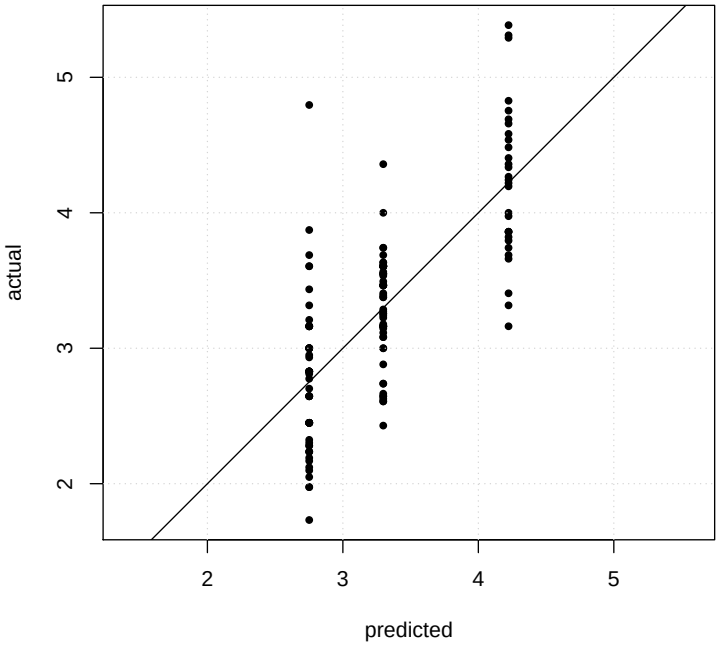
Predict back at calibration points:

```
> p.rpp <- predict(tree.p, newdata=obs)
> length(unique(p.rpp))

[1] 3

> summary(r.rpp <- obs$sqrtCEC1 - p.rpp)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.0600 -0.3020  0.0234  0.0000  0.2870  2.0400

> sqrt(sum(r.rpp^2)/length(r.rpp))

[1] 0.48413
```

Here we see the fitting errors.

1:1 plot: actual vs. fits

```
> summary(r.rpart <- obs$sqrtCEC1 - p.rpp)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.0600 -0.3020  0.0234  0.0000  0.2870  2.0400

> sqrt(sum(r.rpart^2)/length(r.rpart))

[1] 0.48413

> plot(obs$sqrtCEC1 ~ p.rpp, asp=1, pch=20, xlab="predicted", ylab="actual"); grid(); abline(0,1)
```



Note only three predictions ("rectangles").

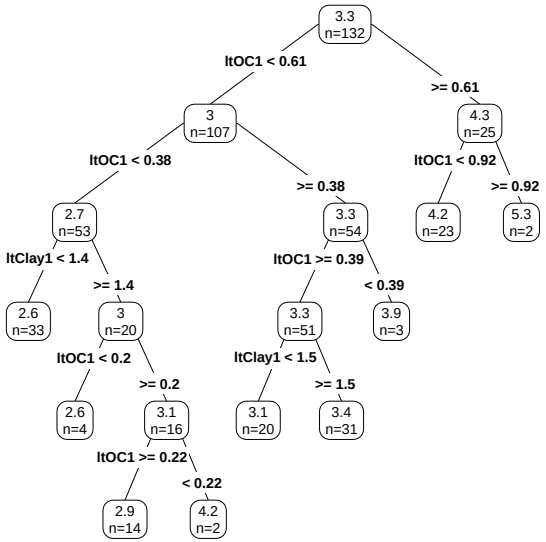
Instablity of regression trees

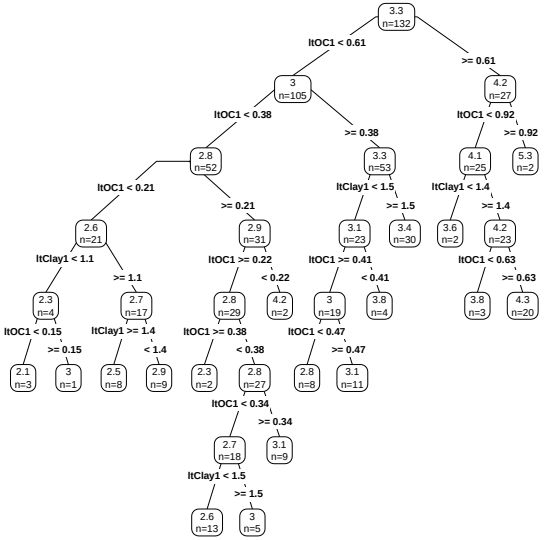
Build several trees with a 90% subset of the observations:

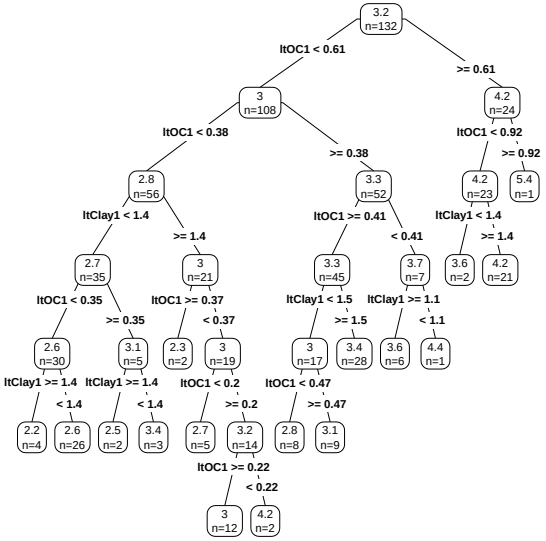
```
> dim(obs)

[1] 147  18

> n <- dim(obs)[1]
> obs.subset <- obs[sample(1:n, size=n*.9),c("sqrtCEC1","ltOC1","ltClay1")]
> dim(obs.subset)  # 10% of observations randomly removed

[1] 132   3

> tree.1 <- rpart(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs.subset, xval=20, minsplit=4, cp=0.0075)
> obs.subset <- obs[sample(1:n, size=n*.9),c("sqrtCEC1","ltOC1","ltClay1")]
> tree.2 <- rpart(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs.subset, xval=20, minsplit=4, cp=0.0075)
> obs.subset <- obs[sample(1:n, size=n*.9),c("sqrtCEC1","ltOC1","ltClay1")]
> tree.3 <- rpart(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs.subset, xval=20, minsplit=4, cp=0.0075)
> obs.subset <- obs[sample(1:n, size=n*.9),c("sqrtCEC1","ltOC1","ltClay1")]
> tree.4 <- rpart(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs.subset, xval=20, minsplit=4, cp=0.0075)
```

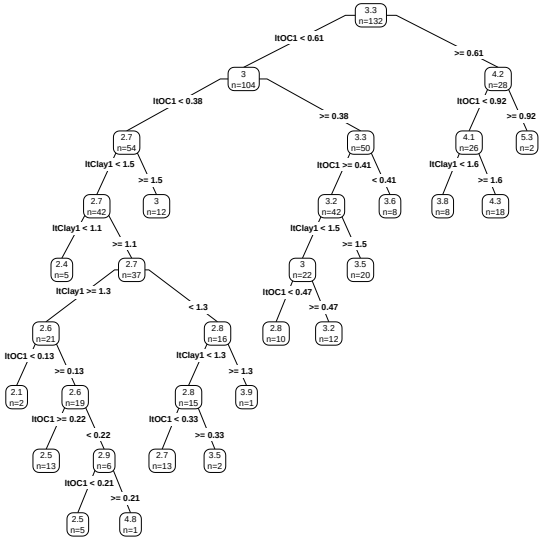
See trees on next page.

Instablity of regression trees – result









Random forests

Problems with regression trees:

1. A small change in the sample set (e.g., a missing or erroneous observation) can make a large change in the tree;
2. Sub-optimal splits propagate down the tree (there is no way to backtrack);
3. Correlated predictors are only used one way;
4. Discontinuous predictions ("rectangles");
5. Different cross-validation splits suggest different complexity parameters for smoothing.

Solution: why one tree when you can have a **forest**?

Procedure

1. Build a **large number of regression trees**, independently, using **different** sets of observations.

2. These are built by **sampling with replacement** from the actual observations.
   - This is sometimes called **bagging**: some observations are **"in the bag"** (used to build the tree) and others **"out of bag"** (used to assess prediction error, see below).
   - Note! this assumes that the sample fairly represents the population!

3. At each split, **randomly** select a predictor.

4. Save all these trees; when predicting, use all of them and **average their predictions**.

5. For each tree we can use observations that were not used to construct it for true **validation**, called **out-of-bag** validation. This gives a good idea of the true prediction error.

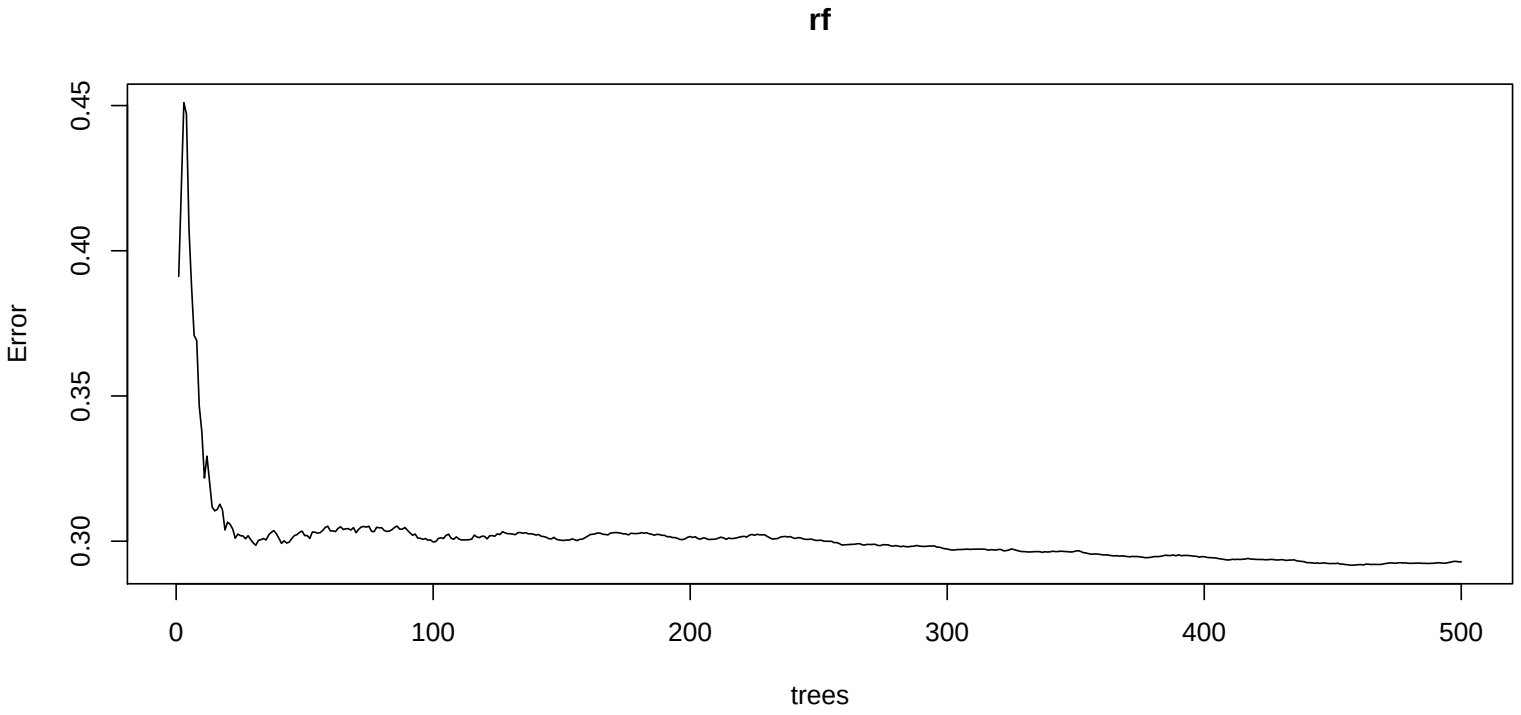
A random forest for the Cameroon CEC vs. OM and clay

```
> library(randomForest)
> rf <- randomForest(sqrtCEC1 ~ ltOC1 + ltClay1, data=obs,
+                    importance=T, na.action=na.omit, mtry=2)
> print(rf)

Call:
 randomForest(formula = sqrtCEC1 ~ ltOC1 + ltClay1, data = obs,      importance = T, mtry = 2, na.action
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 0.2929
                    % Var explained: 44.52

> importance(rf)

        %IncMSE IncNodePurity
ltOC1   45.1537        57.580
ltClay1  2.9915        13.392
```

- %IncMSE percent increase in mean squared error if the variable is not used
- IncNodePurity increase in node purity (reduction in within-node variance) if the variable is used

How many trees are needed to make a forest?

```
> plot(rf)
```



Each run is different (due to randomness); about 250 seem to be adequate in this case (too much fluctuation with fewer trees, very little improvement with more).

No need to prune, the different trees average out the noise.

Prediction with a random forest

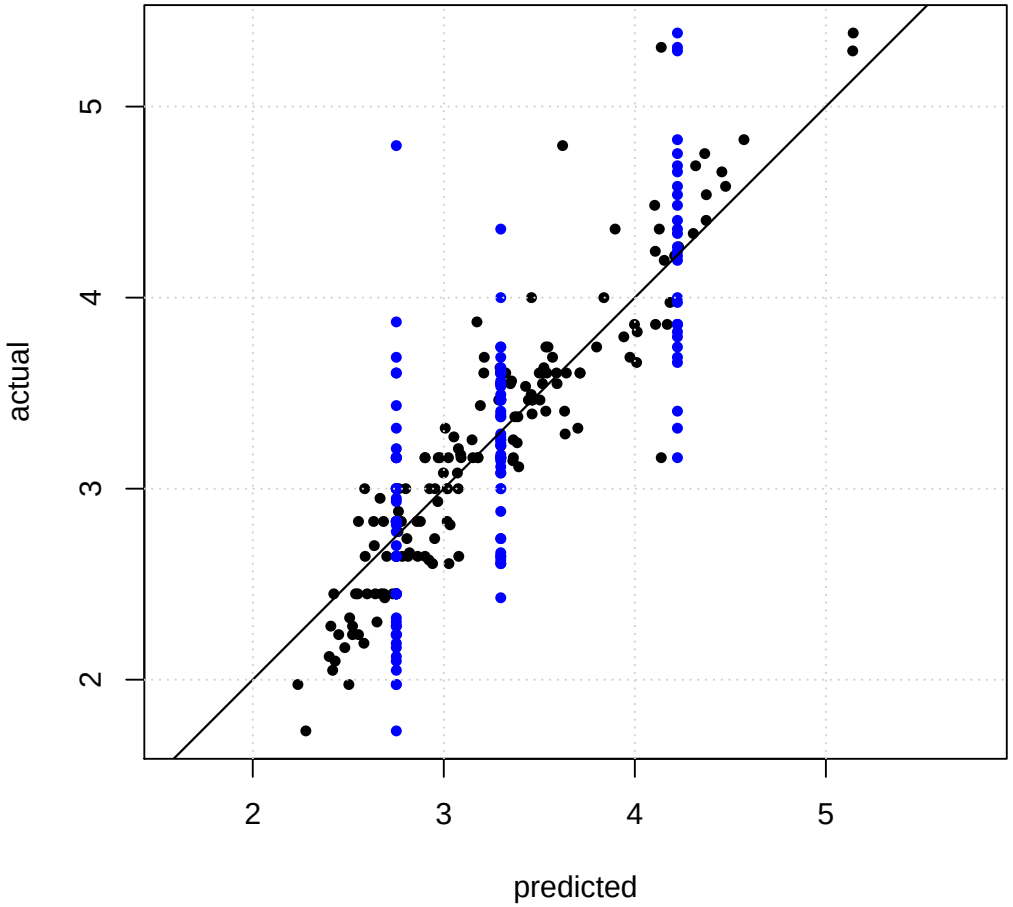
Predict back at calibration points:

```
> p.rf <- predict(rf, newdata=obs)
> length(unique(p.rf))

[1] 137

> summary(r.rf <- obs$sqrtCEC1 - p.rf)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.97600 -0.19700  0.00049 -0.00246  0.14800  1.17000

> sqrt(sum(r.rf^2)/length(r.rf))

[1] 0.27683
```

Note much lower calibration RMSE than from the single regression tree.

1:1 plot: actual vs. fits: random forest and single regression tree

```
> plot(obs$sqrtCEC1 ~ p.rf, asp=1, pch=20, xlab="predicted", ylab="actual")
> points(obs$sqrtCEC1 ~ p.rpp, asp=1, pch=20, col="blue"); grid(); abline(0,1)
> abline(0,1); grid()
```



Out-of-bag validation

The **out-of-bag** validation summarizes the predictions at observations that were omitted in each of the trees in the forest.

```
> r.rf.oob <- predict(rf)
> sqrt(sum(r.rf.oob^2)/length(r.rf.oob))

[1] 3.3277
```

This is a much higher error than the calibration error:

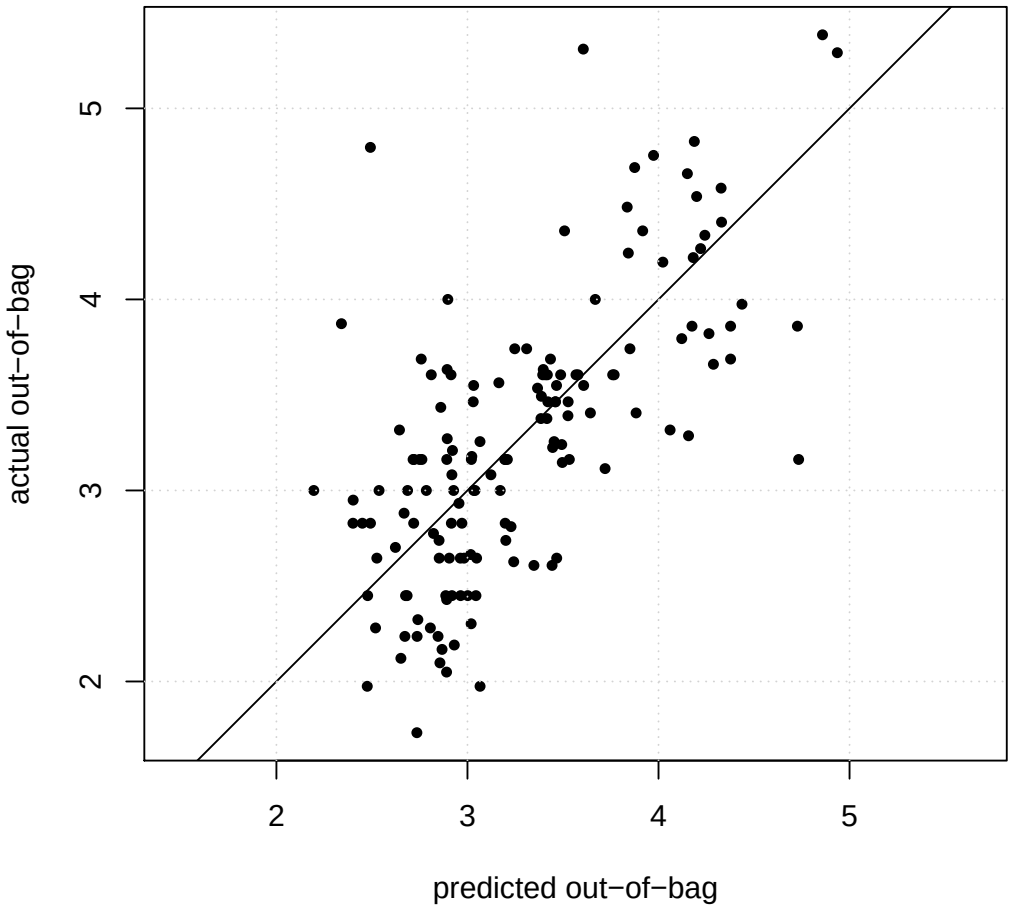
- Calibration: 0.28 $\sqrt{\text{cmol}^+ \text{ (kg soil)}^{-1}}$
- Out-of-bag: 3.33 $\sqrt{\text{cmol}^+ \text{ (kg soil)}^{-1}}$

This is a realistic estimate of the prediction error, if applied to new observations.

We see this graphically on the next page.

1:1 plot actual vs. out-of-bag prediction

```
> plot(obs$sqrtCEC1 ~ r.rf.oob, asp=1, pch=20, xlab="predicted out-of-bag", ylab="actual out-of-bag")
> abline(0,1); grid()
```



Topic: Factor Analysis

Here we consider the **inter-relations** between a set of variables

- Often the set of **predictors** which might be used in a multiple linear regression.

This is an analysis of the **structure** of the **multivariate feature space** covered by a set of variables.

Uses:

1. Discover relations between variables, and possible **groupings**
2. Diagnose multi-collinearity;
3. Identify **representative** variables, e.g., for a minimum data set to be used in regression;
4. Define **synthetic variables** to be used directly in regression.

Principal Components Analysis (PCA)

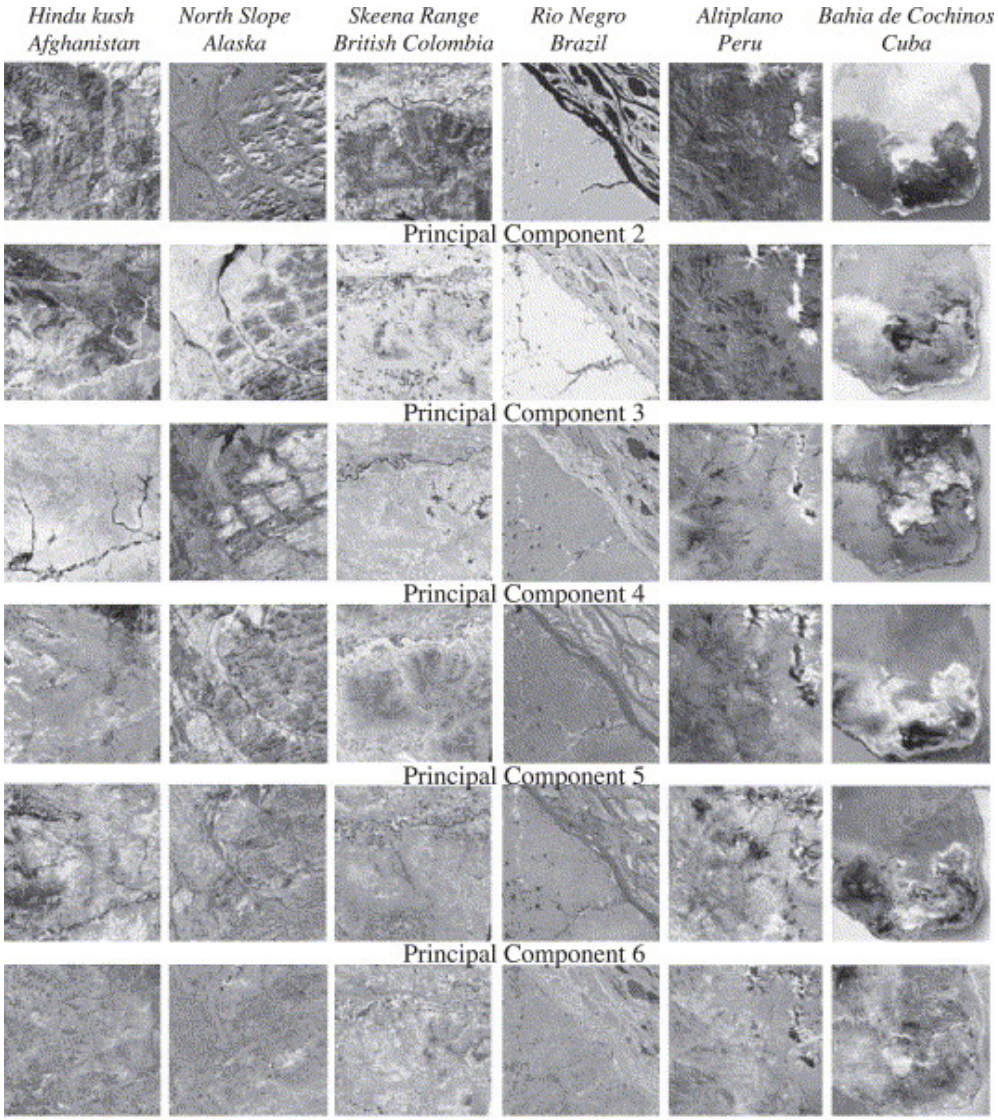
The simplest form of factor analysis; it is a multivariate **data reduction** technique.

- The **vector space** made up of the original variables is **projected** onto another space;
- The new space has the **same dimensionality** as the original[1], i.e., there are as many variables in the new space as in the old;
- In this space the new **synthetic variables**, also called **principal components** are **orthogonal** to each other, i.e. completely uncorrelated;
- The synthetic variables are arranged in **decreasing order of variance explained**.

These synthetic variables can often be **interpreted** by the analyst, that is, they represent some composite attribute of the objects of study.

[1] unless the original was rank-deficient

Visualize: (1) uncorrelated; (2) decreasing information content



Source: Small, C. (2004). The Landsat ETM+ spectral mixing space. *Remote Sensing of Environment*, 93, 1-17

Standardized or not

Two forms:

Standardized each variable has its mean subtracted (so $\overline{x_{\cdot j}} = 0$) and is divided by its sample standard deviation (so $\sigma(x_{\cdot j}) = 1$);

- All variables are equally important, no matter their absolute values or spreads;
- This is usually what we want.

Unstandardized use the original variables, in their original scales of measurement; generally the means are also subtracted to centre the variables

- Variables with larger absolute values and wider spreads are more important, since they contribute more to the original variance

Example: Cameroon soil properties

```
> # non-standardized
> summary(pc <- prcomp(obs[,c("CEC1","Clay1","OC1")]))

Importance of components:
                          PC1   PC2     PC3
Standard deviation     14.282 4.192 0.93299
Proportion of Variance  0.917 0.079 0.00391
Cumulative Proportion   0.917 0.996 1.00000

> # standardized
> summary(pc.s <- prcomp(obs[,c("CEC1","Clay1","OC1")], scale=TRUE))

Importance of components:
                          PC1   PC2    PC3
Standard deviation     1.506 0.690 0.5044
Proportion of Variance 0.756 0.159 0.0848
Cumulative Proportion  0.756 0.915 1.0000
```

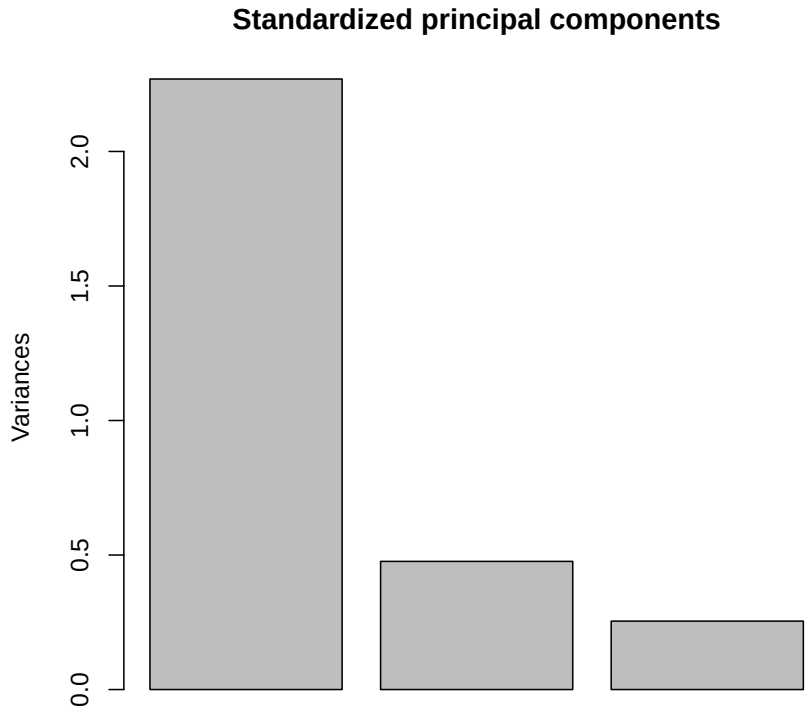
Interpretation

- **Proportion** of variance explained by component
  - always decreasing;
  - here, first component explains most of total variation
- **Cumulative proportion** for components to that number
  - always increasing, ends at 100% explained
- **Standardization** tends to lower the proportion in the first few components; it avoids the PCs being dominated by the numerically-larger variables.

Screeplot

A simple visualization of the variance explained.

```
> screeplot(pc.s, main = "Standardized principal components")
```



Rotations

The synthetic variables are composed of a linear combination of the originals; this is a **rotation** of the axe by the eigenvectors, also called the **loadings** of each original variable:

```
> pc.s$rotation

             PC1       PC2        PC3
CEC1  -0.58910   0.45705 -0.666384
Clay1 -0.54146 -0.83542 -0.094322
OC1   -0.59982   0.30525  0.739619
```

Interpretation (note: signs are arbitrary, depend on algorithm used):

PC1 overall magnitude, "soil activity"; all three original variables contribute about equally and in the same direction; about 76% of the variance;

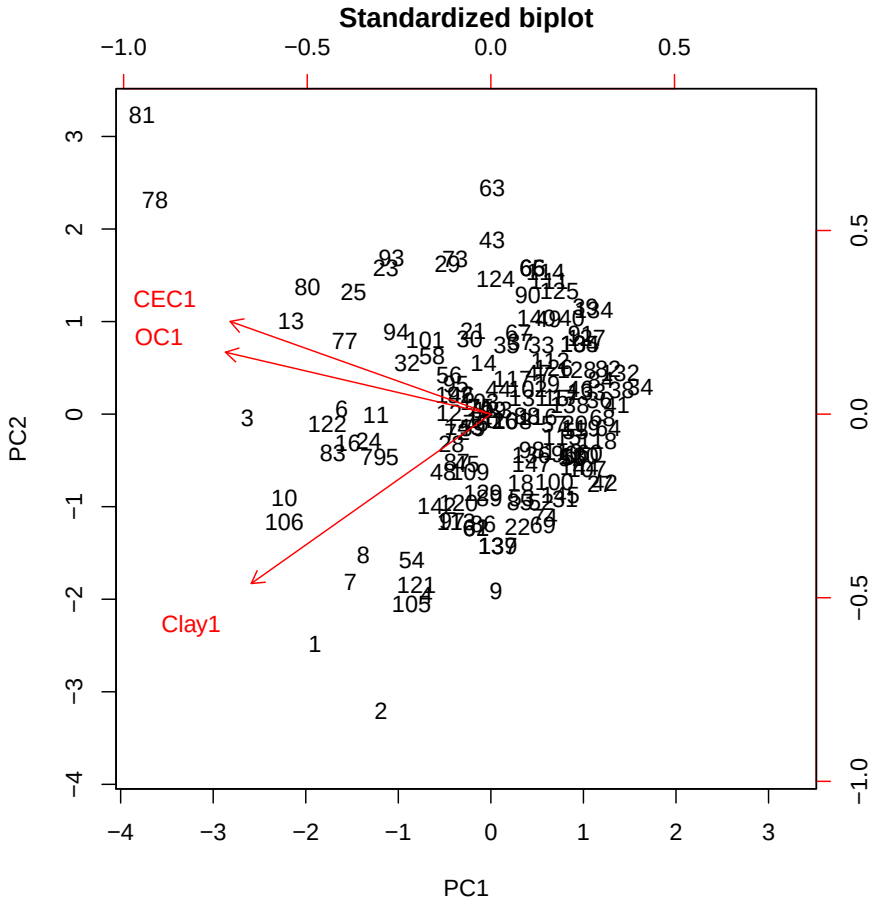
PC2 contrast between clay and (CEC and OC); soils with high clay but relatively low CEC and OC, or vice-verse; about 16% of the variance;

PC3 contrast between clay and CEC; about 8% of the variance.

Biplots

These show positions of the observations as synthetic variables (bottom, left axes) and the correlations/variances of the original standardized variables (top, right axes):

```
> biplot(pc.s, main = "Standardized biplot", pc.biplot = TRUE)
```



Interpretation of biplots

- **Length** of vector is variance explained in this plane;
- **Angle** between vectors is degree of correlation (closer = more correlated);
- Individual observations are plotted with their PC **scores** (values in the PC space);
- Points close in this space have similar properties with respect to these two PCs.

Retrieving synthetic variables

Also called the "scores".

These can be returned from PCA and then used in any analysis.

```
> pc.s <- prcomp(obs[, c("CEC1", "Clay1", "OC1")], scale = TRUE,
+     retx = TRUE)
> summary(pc.s$x)

      PC1               PC2               PC3
 Min.   :-5.677   Min.   :-2.213   Min.   :-2.165
 1st Qu.:-0.634   1st Qu.:-0.399   1st Qu.:-0.266
 Median : 0.228   Median :-0.019   Median :-0.018
 Mean   : 0.000   Mean   : 0.000   Mean   : 0.000
 3rd Qu.: 1.145   3rd Qu.: 0.415   3rd Qu.: 0.312
 Max.   : 2.434   Max.   : 2.234   Max.   : 1.603
```

These are now variables ready to use in regression models.

PCs are uncorrelated

Proof that the PCs are uncorrelated (as opposed to the original variables):

```
> # PCs
> round(cor(pc.s$x),5)

    PC1 PC2 PC3
PC1   1   0   0
PC2   0   1   0
PC3   0   0   1

> # original variables
> round(cor(obs[,c("CEC1","Clay1","OC1")]),5)

         CEC1   Clay1     OC1
CEC1  1.00000 0.55796 0.74294
Clay1 0.55796 1.00000 0.59780
OC1   0.74294 0.59780 1.00000
```

Mathematics

PCA is a direct calculation from a data matrix. The key insight is that the eigen decomposition automatically orders the synthetic variables into descending amounts of variance (predictive power), and ensures they are orthogonal.

This was worked out by Hotelling in 1933.

$\mathbf{X}$: scaled and centred data matrix: rows are observations, columns are variables measured at each observation; centred and scaled per column

$\mathbf{C} = \mathbf{X}^T\mathbf{X}$: the correlation matrix; this is symmetric and positive-definite (all real roots)

$|\mathbf{C} - \lambda\mathbf{I}| = 0$: a determinant to find the **characteristic values**, also called **eigenvalues**, of the correlation matrix.

Then the axes of the new space, the **eigenvectors** γ_j (one per dimension) are the solutions to $(\mathbf{C} - \lambda_j I)\gamma_j = \mathbf{0}$

Obtain synthetic variables by projection: $\mathbf{Y} = \mathbf{PX}$ where $\mathbf{P}$ is the row-wise eigenvectors (rotations).

Details

In practice the system is solved by the Singular Value Decomposition (SVD) of the data matrix, for numerical stability.

This is equivalent but more stable than directly extracting the eigenvectors of the correlation matrix.

Accessible explanations:

- Davis, J. C. (2002). *Statistics and data analysis in geology*. New York: John Wiley & Sons.

- Legendre, P., & Legendre, L. (1998). *Numerical ecology*. Oxford: Elsevier Science.

Topic: Linear model for categorical predictors

Predictors may be **categorical**:

- **Nominal**: unordered categories
- **Ordinal**: categories with a natural order but *not* on an interval scale

These can also be modelled with the linear model $y = BX + \varepsilon$.

Example dataset

Tropenbos Cameroon research soil profiles

Categorical predictors:

- 4 agro-ecological **zones**
- 8 **previous landuses**
- 3 **soil groups** in the World Reference Base for Soil Classification

Summary statistics

Zone:

```
zone
 1  2  3  4
 8 40 63 36
```

Previous land cover:

```
LC
  BF   CF   FF   FV  MCA  OCA YANA  YOP
  19   15   17   69   11   14    1    1
```

Soil groups:

```
wrb1
  1   2   3
 40   3 104
```

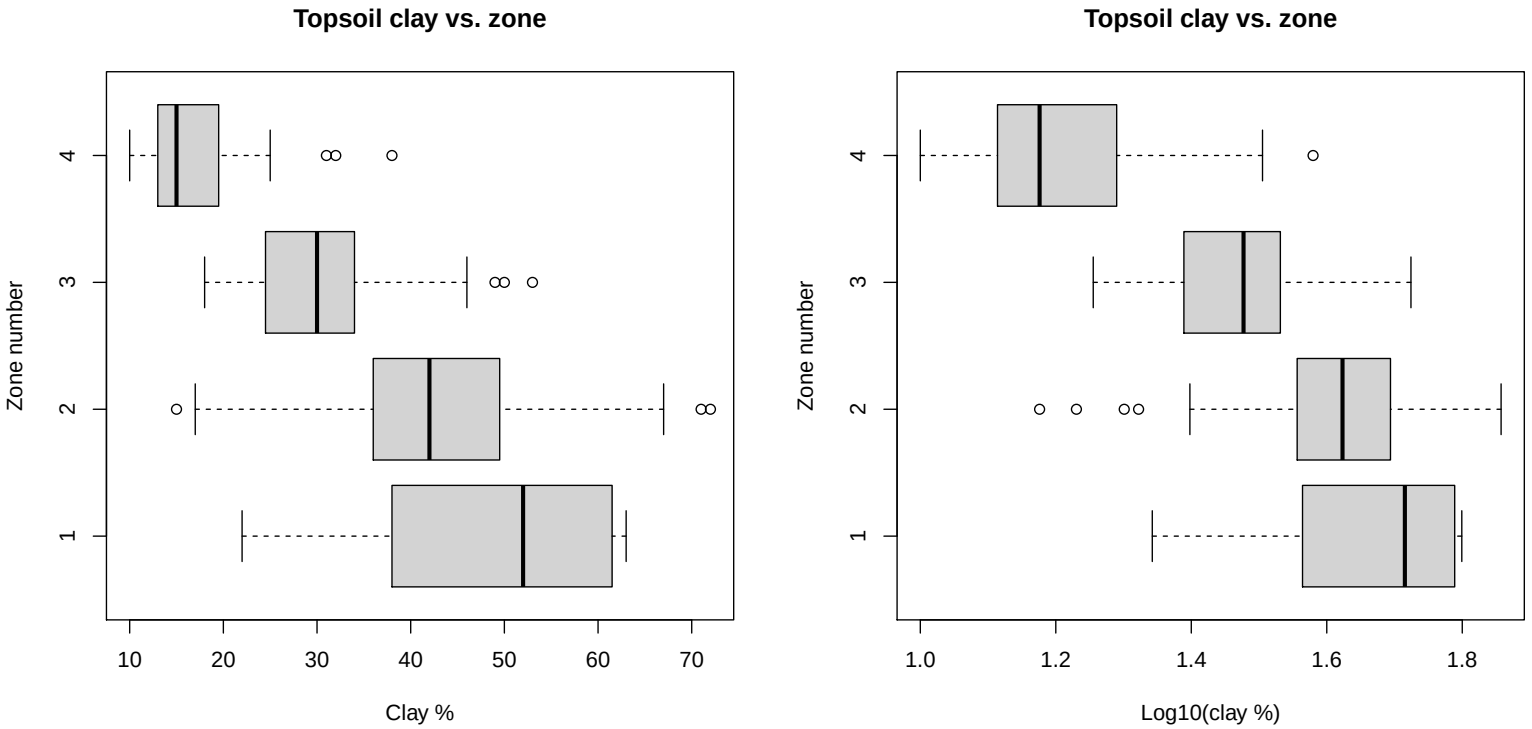
Model from a single categorical predictor

Research question: do the different **zones** (represented by villages) have different **soil properties**?

Example: topsoil clay content (log-transformed)

Visualizing differences in response by category

Untransformed (left) and log10-transformed (right)



Boxplots show **median**, **1st and 3rd quartiles** (box limits), **fences** (1.5 x Inter-Quartile Range away from quartiles), and **boxplot outliers**

Linear model: differences in response by category

Rows of the **design matrix** X have a single 1 corresponding to the zone of the observation, 0 for the others.

```
    (Intercept) zone2 zone3 zone4 observation.zone
1             1     1     0     0                2
2             1     1     0     0                2
3             1     0     0     0                1
4             1     0     0     0                1
5             1     1     0     0                2

     (Intercept) zone2 zone3 zone4 observation.zone
143            1     1     0     0                2
144            1     1     0     0                2
145            1     1     0     0                2
146            1     0     1     0                3
147            1     0     1     0                3
```

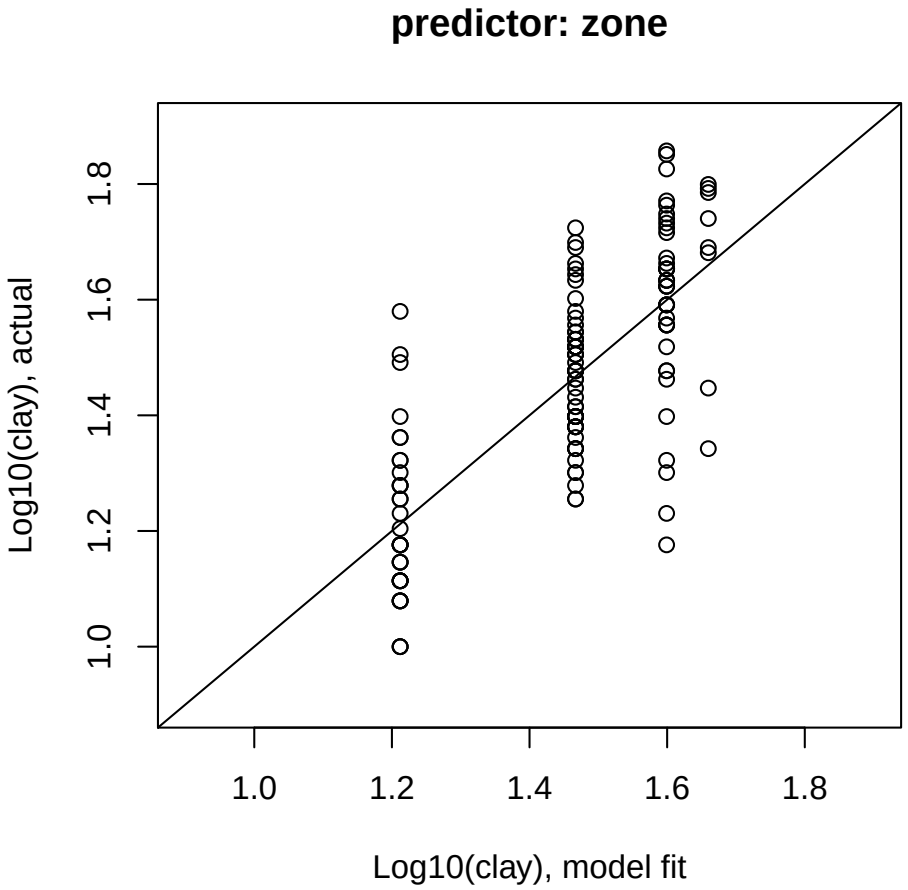
Model summary

```
Call:
lm(formula = ltClay1 ~ zone)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4231 -0.0866  0.0103  0.0698  0.3678

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.6598     0.0477   34.77  < 2e-16
zone2        -0.0606     0.0523   -1.16  0.24851
zone3        -0.1930     0.0507   -3.81  0.00021
zone4        -0.4479     0.0528   -8.49  2.5e-14

Residual standard error: 0.135 on 143 degrees of freedom
Multiple R-squared: 0.559,        Adjusted R-squared: 0.549
F-statistic: 60.4 on 3 and 143 DF,  p-value: <2e-16
```

About half (0.549) of the variability in log10-topsoil clay is explained by the zone in which the observation was made.

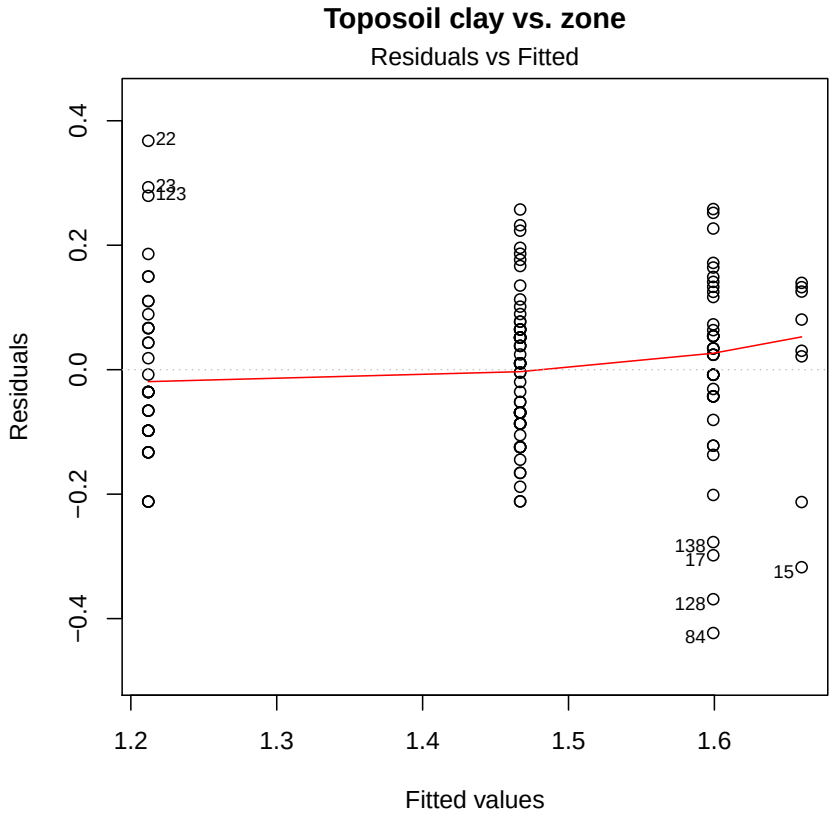
Zones 3 and 4 have significantly lower clay contents, on average, than Zone 1. Zone 2 is lower but not significantly so.

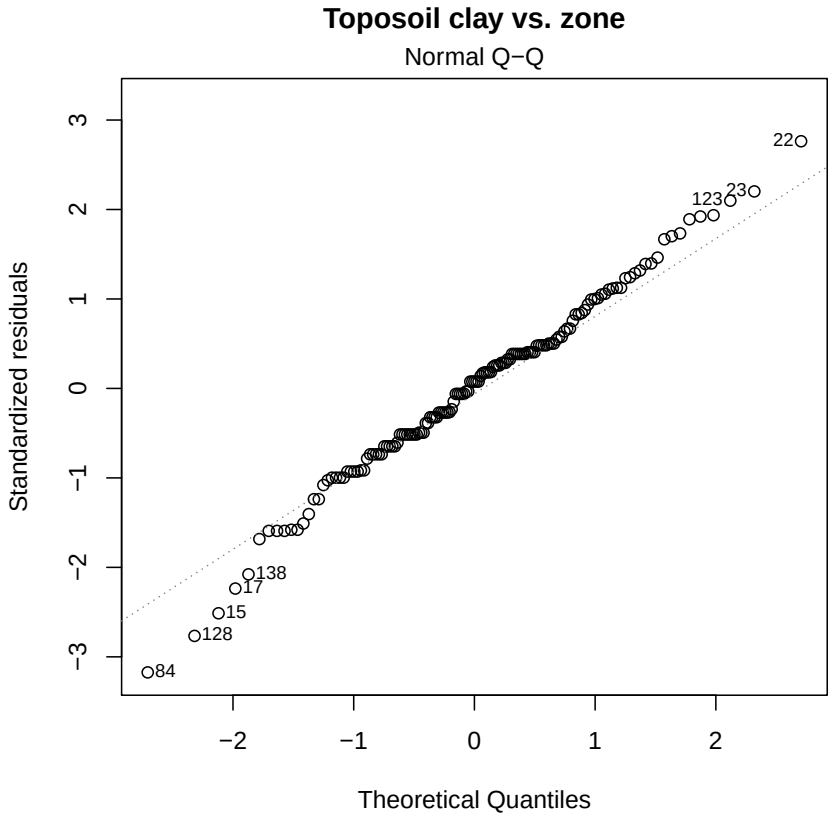
Linear model: Actual vs. fits



Note only one prediction per class.

Linear model: Regression diagnostics

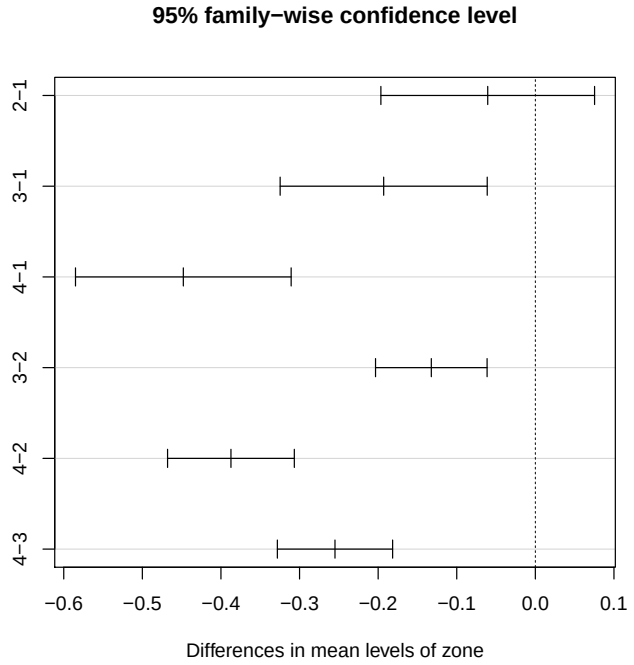




Differences between class means

Using Tukey's "Honestly-significant difference" (HSD) test at the default 95% confidence level:

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = lmclay.zone)

$zone
         diff      lwr       upr   p adj
2-1 -0.060591 -0.19652  0.075342 0.65379
3-1 -0.192955 -0.32469 -0.061222 0.00118
4-1 -0.447866 -0.58505 -0.310680 0.00000
3-2 -0.132364 -0.20332 -0.061407 0.00002
4-2 -0.387275 -0.46791 -0.306644 0.00000
4-3 -0.254911 -0.32824 -0.181582 0.00000
```



Topic: Mixed models

It is possible to mix both **continuous** and **categorical** predictors in one model.

This is a form of **multiple linear regression**

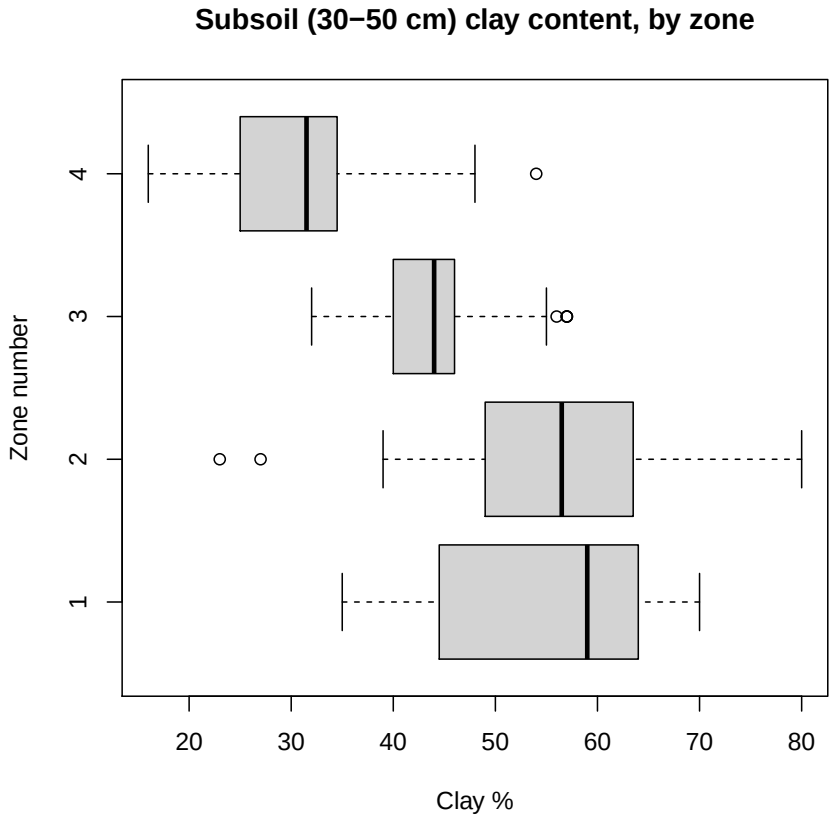
The linear model form $\mathbf{y} = \mathbf{BX} + \varepsilon$ is applicable.

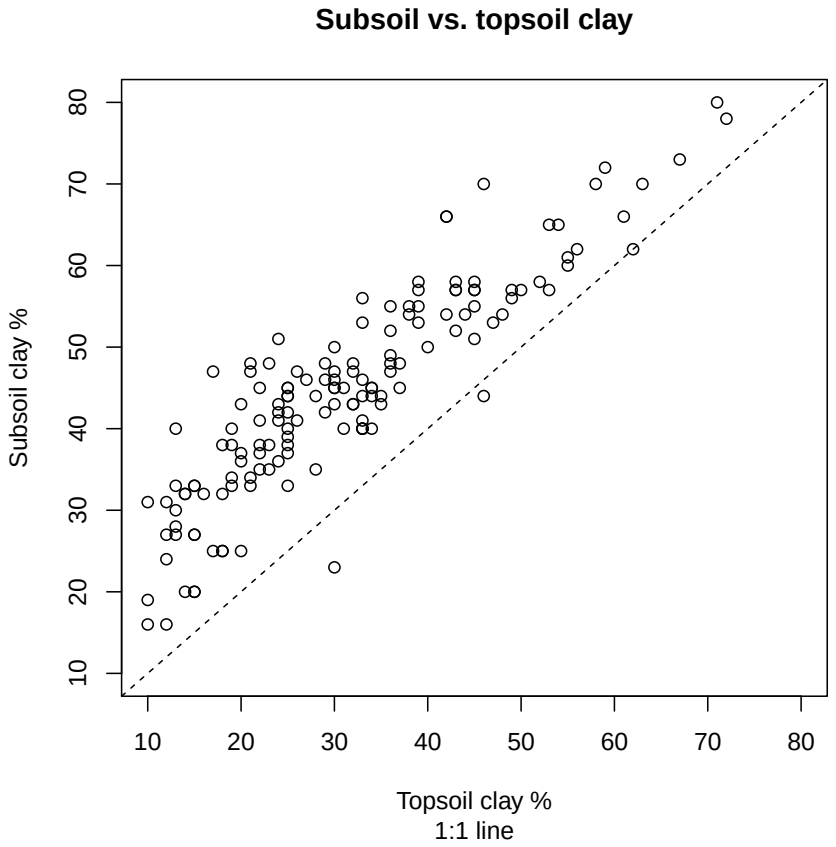
A simple mixed model

Objective: to predict the **subsoil** clay content (30–50 cm depth) from the **topsoil** clay content (0–10 cm depth) and/or **zone**.

Purpose: avoid expensive / laborious augering to 50 cm and extra lab. work

Visualizing the single predictors





Fairly equal spread per zone

Subsoil almost always has more clay than the topsoil (agrees with theory of soil formation in this zone).

Single-predictor models

(1) Subsoil clay vs. topsoil clay (continuous predictor):

```
Call:
lm(formula = Clay5 ~ Clay1)

Residuals:
    Min      1Q  Median      3Q     Max
-20.626  -3.191   0.005   3.387  14.150

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.7586     1.1556    16.2   <2e-16
Clay1         0.8289     0.0338    24.5   <2e-16

Residual standard error: 5.69 on 145 degrees of freedom
Multiple R-squared: 0.806,        Adjusted R-squared: 0.805
F-statistic:  602 on 1 and 145 DF,  p-value: <2e-16
```

(continued . . .)

Single-predictor models

(2) Subsoil clay vs. zone (categorical predictor):

```
Call:
lm(formula = Clay5 ~ zone)

Residuals:
   Min     1Q Median     3Q    Max
-32.95  -5.40   0.16   3.16  24.05

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    55.00       3.21   17.14  < 2e-16
zone2           0.95       3.52    0.27   0.7874
zone3         -11.16       3.41   -3.28   0.0013
zone4         -23.67       3.55   -6.67  5.2e-10

Residual standard error: 9.08 on 143 degrees of freedom
Multiple R-squared: 0.513,        Adjusted R-squared: 0.502
F-statistic: 50.1 on 3 and 143 DF,  p-value: <2e-16
```

Design matrix

Rows of the **design matrix** X have a single 1 corresponding to the zone of the observation, 0 for the others; and the actual value of topsoil log10-clay. The interaction model also has the product.

Additive model:

	(Intercept)	zone2	zone3	zone4	Clay1
1	1	1	0	0	72
2	1	1	0	0	71
3	1	0	0	0	61
4	1	0	0	0	55
5	1	1	0	0	47

Interaction model:

	(Intercept)	zone2	zone3	zone4	Clay1	zone2:Clay1	zone3:Clay1
1	1	1	0	0	72	72	0
2	1	1	0	0	71	71	0
3	1	0	0	0	61	0	0
4	1	0	0	0	55	0	0
5	1	1	0	0	47	47	0

	zone4:Clay1
1	0
2	0
3	0
4	0
5	0

Model summary – additive

```
Call:
lm(formula = Clay5 ~ zone + Clay1)

Residuals:
   Min     1Q Median     3Q    Max
-24.09  -2.99   0.15   3.14  13.89

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  19.3244     2.9054    6.65  5.8e-10
zone2         5.6945     2.1060    2.70   0.0077
zone3         2.2510     2.1831    1.03   0.3043
zone4        -0.6594     2.5365   -0.26   0.7953
Clay1         0.7356     0.0452   16.26  < 2e-16

Residual standard error: 5.39 on 142 degrees of freedom
Multiple R-squared: 0.83,        Adjusted R-squared: 0.825
F-statistic:  173 on 4 and 142 DF,  p-value: <2e-16
```

About four-fifths (0.825) of the variability in subsoil clay is explained by the zone in which the observation was made and the observed topsoil clay content.

Zones 2 is the only one that differs significantly from Zone 1; it has an average of 5.69% more clay.

Model summary – interaction

```
Call:
lm(formula = Clay5 ~ zone * Clay1)

Residuals:
    Min      1Q  Median      3Q     Max
-24.048  -2.883   0.515   2.889  13.233

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.5362     6.4093    2.27    0.025
zone2        10.3477     6.9759    1.48    0.140
zone3        12.2331     6.9145    1.77    0.079
zone4        -1.8272     6.8954   -0.26    0.791
Clay1         0.8343     0.1265    6.59  8.2e-10
zone2:Clay1  -0.0955     0.1411   -0.68    0.500
zone3:Clay1  -0.2703     0.1513   -1.79    0.076
zone4:Clay1   0.2471     0.1877    1.32    0.190

Residual standard error: 5.24 on 139 degrees of freedom
Multiple R-squared: 0.842,        Adjusted R-squared: 0.834
F-statistic:  106 on 7 and 139 DF,  p-value: <2e-16
```

Somewhat more (0.834 vs. 0.825) of the variability in subsoil clay is explained by the interaction model vs. the additive model. The Zone3:Topsoil clay interaction is significant.

Comparing models

```
Analysis of Variance Table

Model 1: Clay5 ~ zone * Clay1
Model 2: Clay5 ~ zone + Clay1
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1    139 3813
2    142 4118 -3      -305 3.7  0.013
```

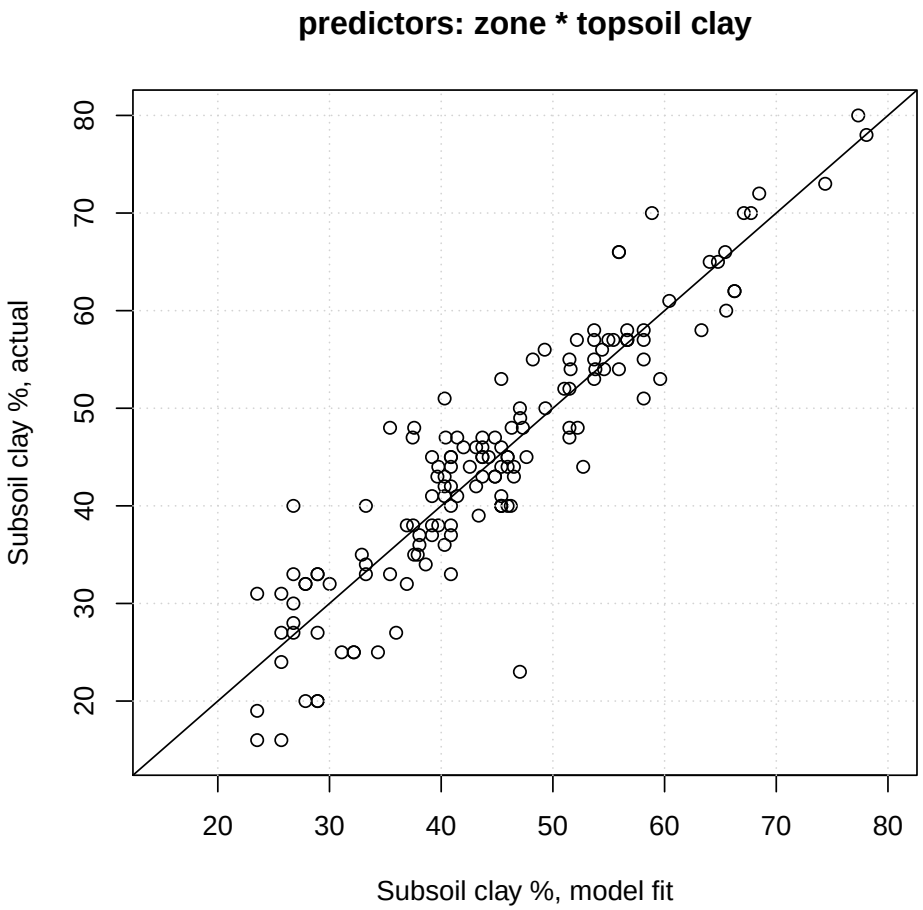
```
Analysis of Variance Table

Model 1: Clay5 ~ zone + Clay1
Model 2: Clay5 ~ zone
  Res.Df   RSS Df Sum of Sq   F Pr(>F)
1    142  4118
2    143 11782 -1     -7664 264 <2e-16
```

```
Analysis of Variance Table

Model 1: Clay5 ~ zone + Clay1
Model 2: Clay5 ~ Clay1
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1    142 4118
2    145 4689 -3      -571 6.57 0.00035
```

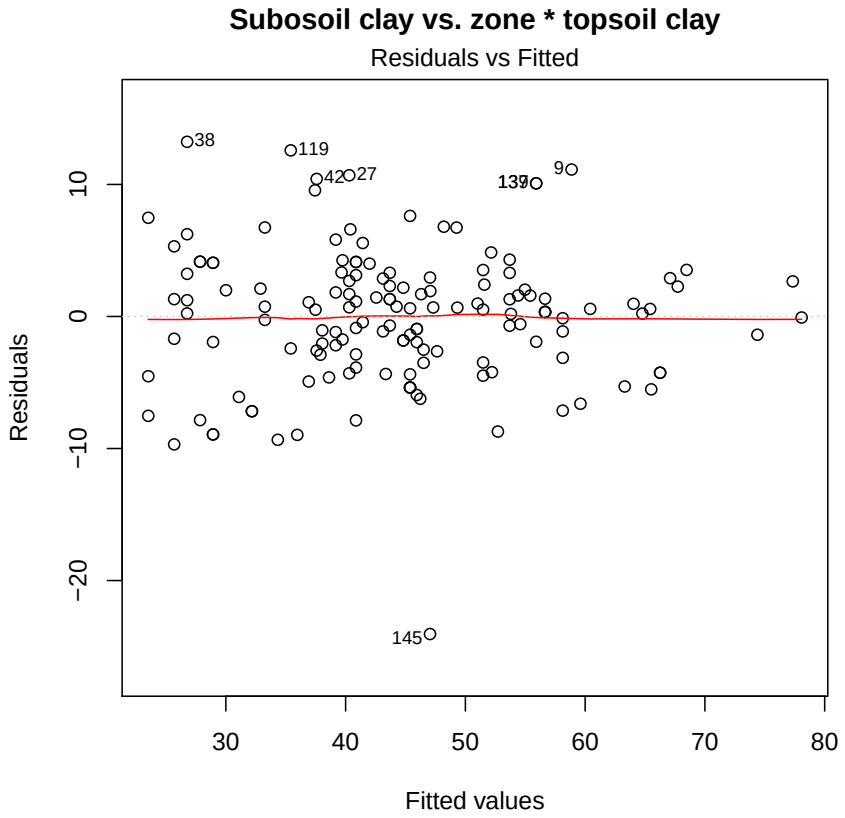
The interaction model is somewhat better than the additive model.

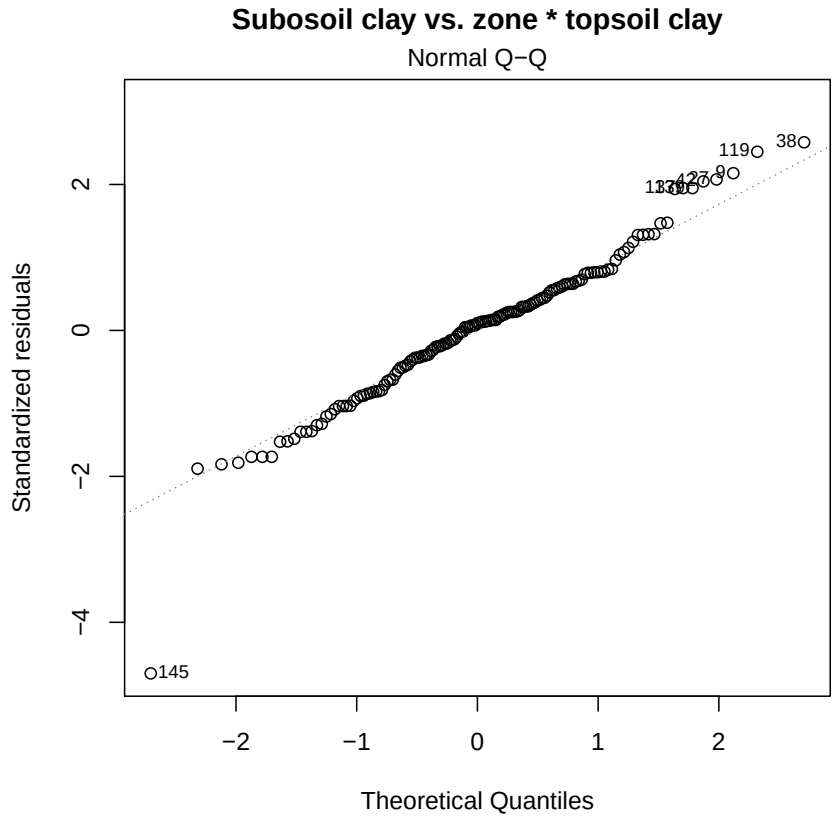
The additive model is much better than the zone-only model, and somewhat better than the topsoil clay-only model.

Interaction mixed model: Actual vs. fits



Interaction mixed model: Regression diagnostics





One very badly-modelled observation! Quite unusual: subsoil clay is well below the topsoil clay. Observational error (mislabelled sample boxes)?

```
[1] "Observation 145: Actual: 23 %; Fitted: 47 %; Located in zone 2 ; topsoil clay: 30 %"
```

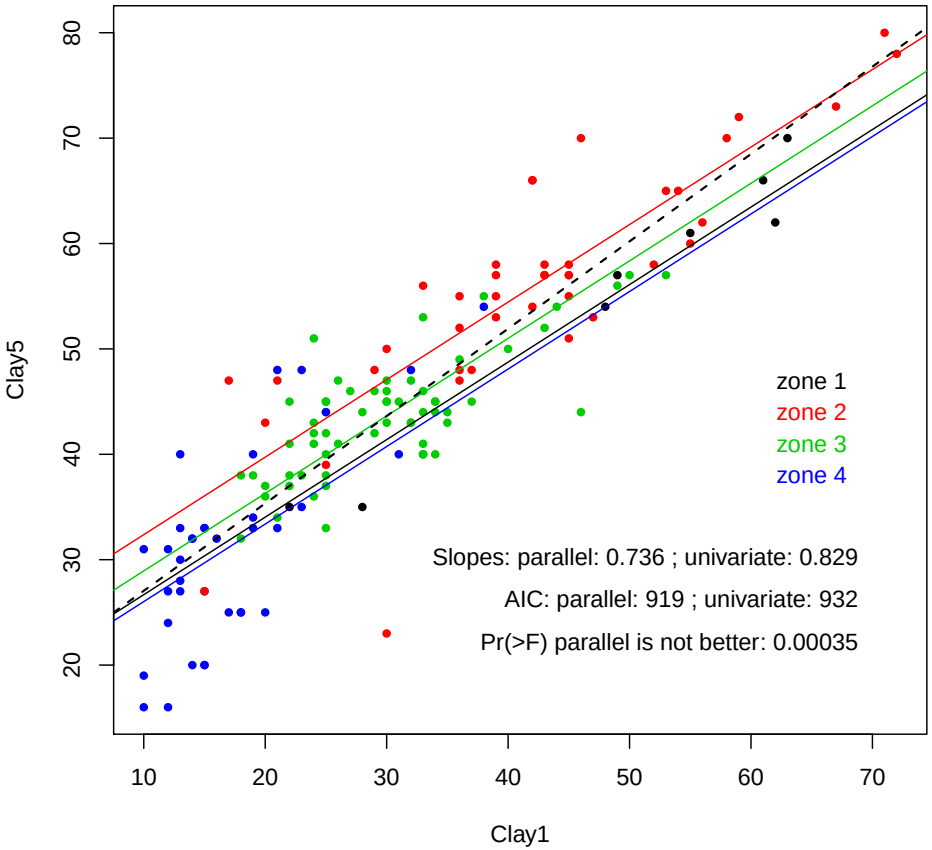
Visualizing the additive model

Parallel regression

- same **slope** on **continuous** predictor
- different **intercepts** per category on **categorical** predictor.

Does *not* allow a different **response** per category, only a different **level**.

Additive model: parallel regression



Clearly the common slope is *not* appropriate for Zone 4.

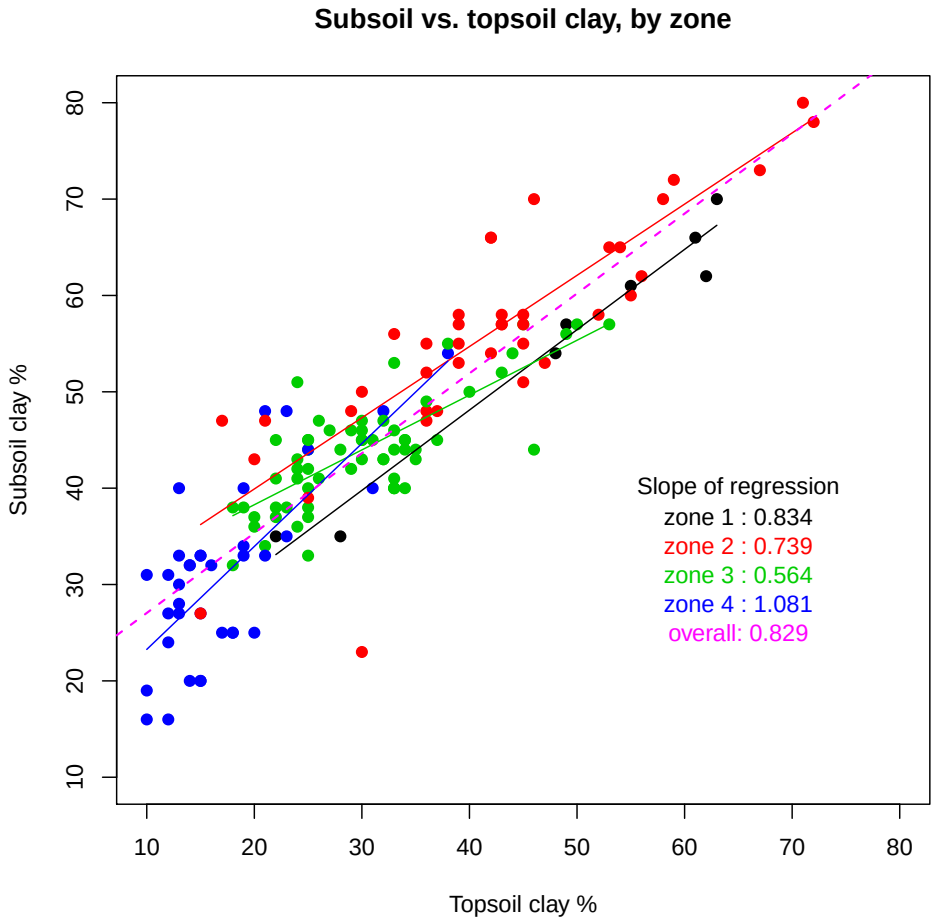
Visualizing the interaction model

Non-parallel regression

- may have different **slopes** on **continuous** predictor, per category
- different **intercepts** per category.

Allows different **responses** per category, and different **levels**.

Interaction model: different slopes per category



Zone 4 has a much steeper slope (and lower overall values); these are low-clay Acrisols, vs. the other zones with medium- to high-clay Ferralsols.

Topic: Robust methods

If the **assumptions** of linear regression are violated, what do we do?

1. Violations of **linearity**: linearize, or **non-linear** methods

2. Residuals not normally-distributed, dependence of residual on fit

   (a) **Non-linearity**: see above
   (b) A few **poorly-modelled** observations; especially **high leverage** (influential): **robust** methods.

3. Variance differs across the range: **heteroscedascity**: **variance-stabilizing transformation**

4. Not a **single relation** through the range: **piecewise** or **local** regression

Robust or **resistant** methods: good performance even if **contamination** from another process.

Robust regression

This fits a regression to the **"good"** observations in a dataset.

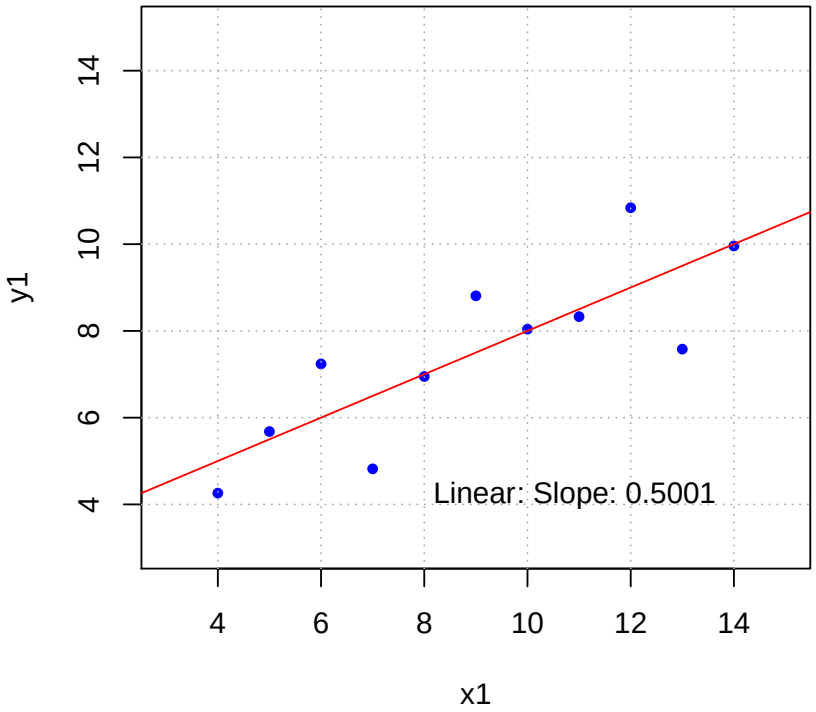
The regression estimator has a high **breakdown** point: how many "bad" points there have to be to distort the equation.

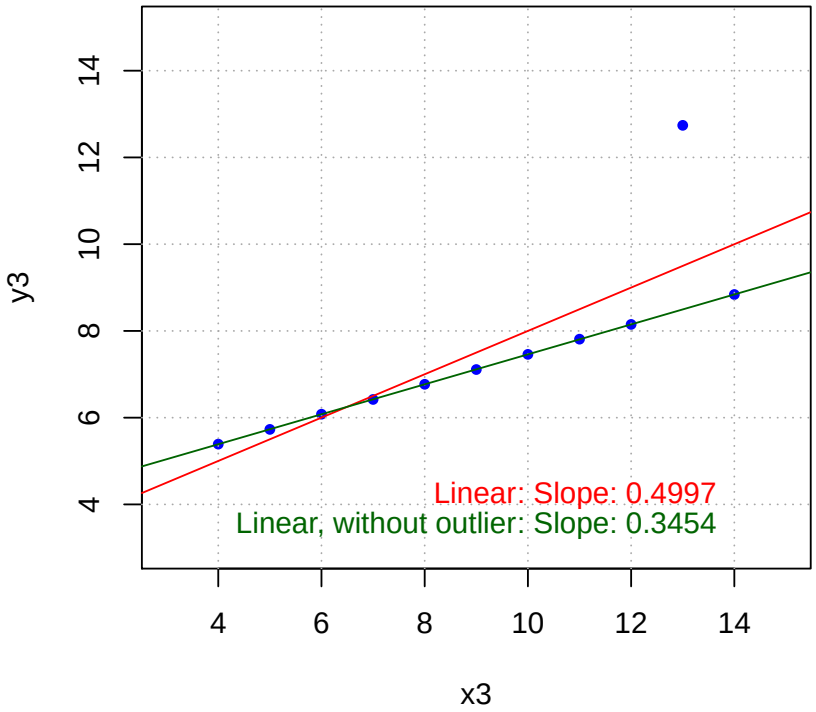
There are many options; here we use the default for the `lqs` function of the `MASS` R package.

Reference: Venables, W. N., & Ripley, B. D. (2002). Modern applied statistics with S (Fourth ed.). New York: Springer-Verlag.

Anscombe example

Compare the noisy-linear with the linear+single outlier Anscombe examples:





(recall: true slope is 0.5)

Robust fit

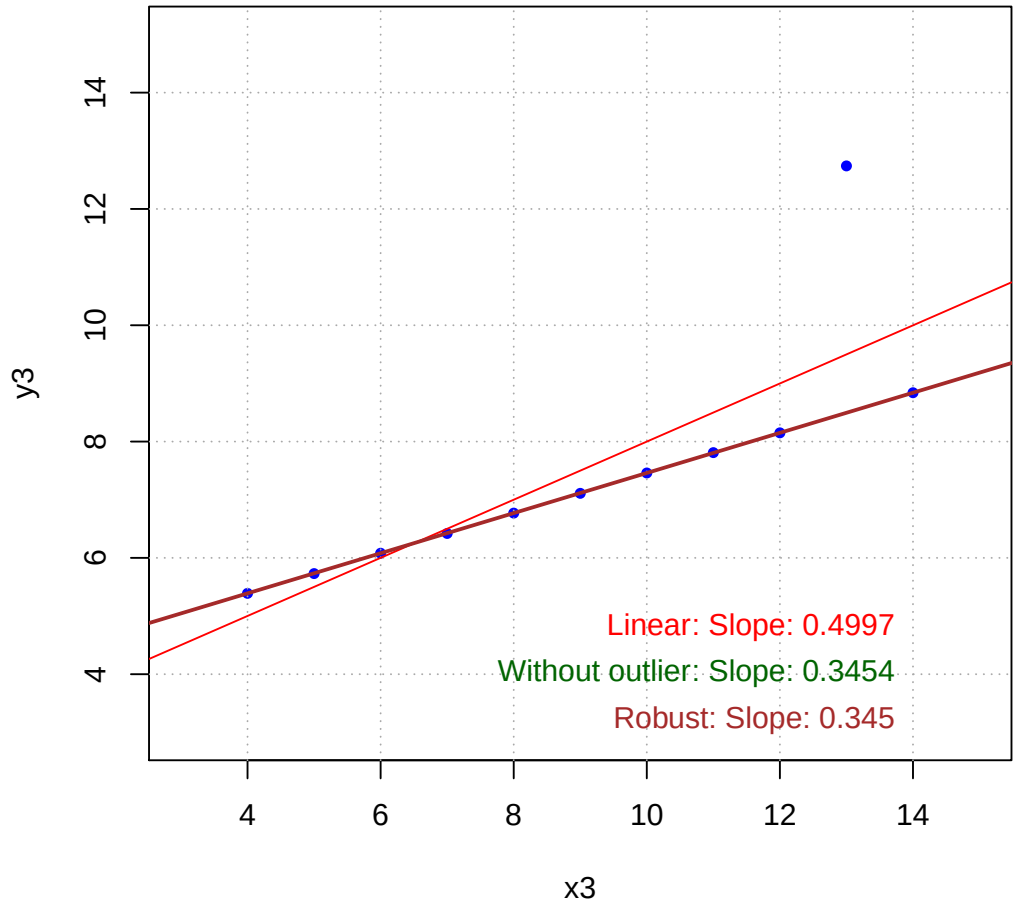
Objective: fit the relation with the outlier automatically.

Minimization criterion: sum of the `floor(n/2) + floor((p+1)/2)` smallest squared residuals (`n` observations, `p` predictors).

```
[1] "Coefficients for least-squares fit:"

(Intercept)          x3
    3.00245     0.49973

[1] "Coefficients for least-squares fit without outlier:"

(Intercept)          x3
    4.00565     0.34539

[1] "Coefficients for resistant fit:"

(Intercept)          x3
      4.010       0.345
```

Note resistant fit very close to fit with only "good" points; automatically more-or-less ignores the outlier.

Visualize robust fit



Local regression

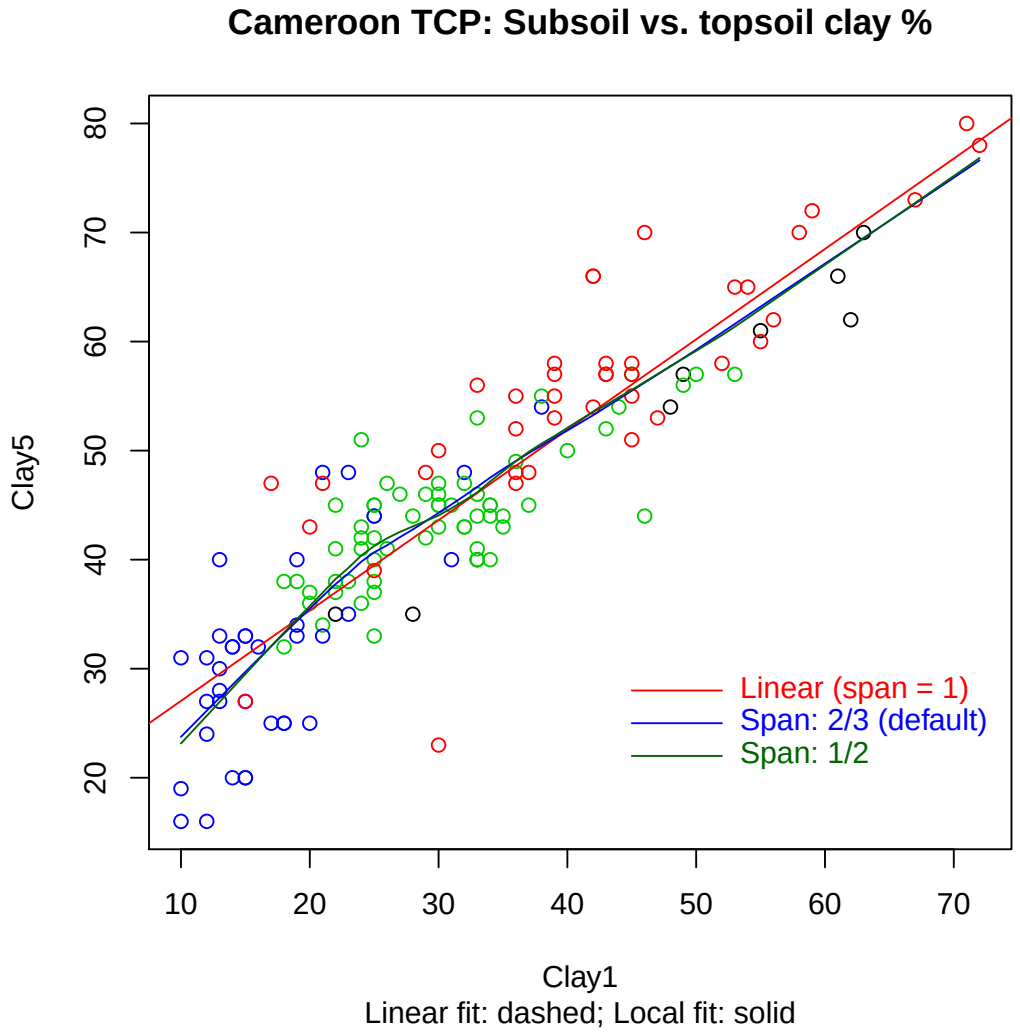
All the methods presented so far assume **one relation** (linear or otherwise) over the entire **range** of the predictor.

Another possibility is **local** regression: fitting in **pieces**.

Many methods, with variable amounts of **smoothing** based on the **span**, i.e. the proportion of the range to consider for each piece.

Here we use the default for the `lowess` function of the R `stats` package, which uses iterated weighted least squares.

Example of local regression



Notice how this adjusts for the high subsoil/topsoil ratios in zone 4 (blue).

Conclusion

Modelling is not simple . . .