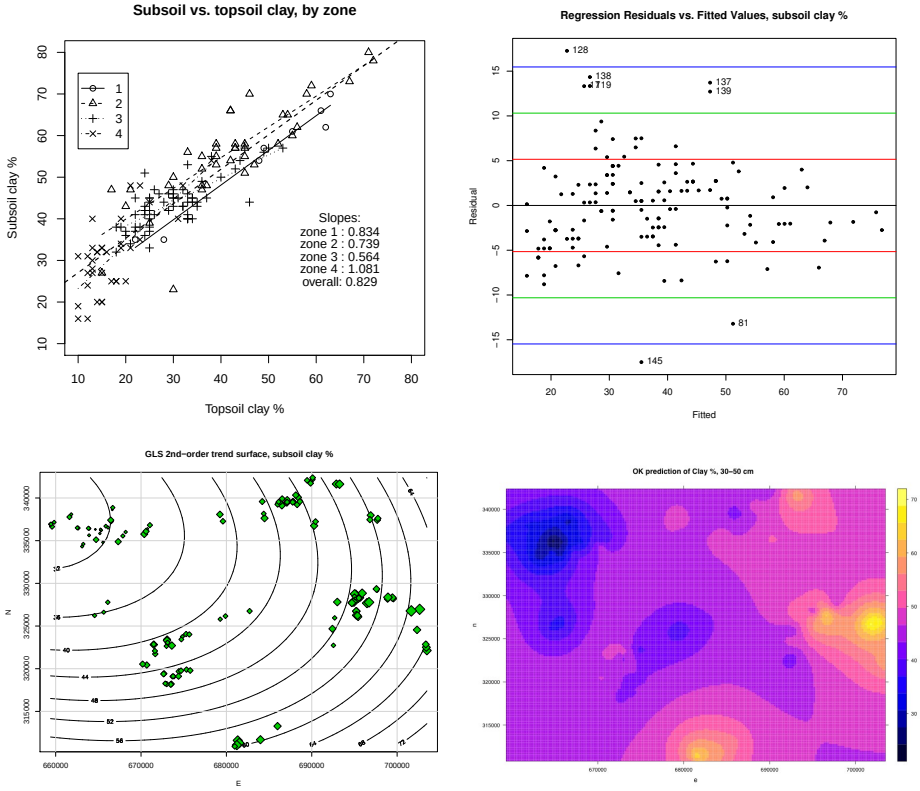
Tutorial:
An example of statistical data analysis using the R environment for statistical computing

D G Rossiter

Version 1.4; May 6, 2017



Contents

1 Introduction **1**

2 Example Data Set **2**
2.1 Loading the dataset . . . 3
2.2 A normalized database structure* . . . 5

3 Research questions **8**

4 Univariarte Analysis **9**
4.1 Univariarte Exploratory Data Analysis . . . 9
4.2 Point estimation; inference of the mean . . . 14
4.3 Answers . . . 15

5 Bivariate correlation and regression **16**
5.1 Conceptual issues in correlation and regression . . . 16
5.2 Bivariate Exploratory Data Analysis . . . 18
5.3 Bivariate Correlation Analysis . . . 22
5.4 Fitting a regression line . . . 23
5.5 Bivariate Linear Regression . . . 25
5.6 Bivariate Regression Analysis from scratch* . . . 28
5.7 Regression diagnostics . . . 30
5.7.1 Fit to observed data . . . 30
5.7.2 Large residuals . . . 31
5.7.3 Distribution of residuals . . . 33
5.7.4 Leverage * . . . 35
5.7.5 DFBETAS * . . . 37
5.8 Prediction . . . 39
5.9 Robust regression* . . . 42
5.10 Structural Analysis* . . . 45
5.11 Structural Analysis by Principal Components* . . . 48
5.12 A more difficult case . . . 49
5.13 Non-parametric correlation . . . 52
5.14 Answers . . . 53

6 One-way Analysis of Variance (ANOVA) **57**
6.1 Exploratory Data Analysis . . . 58
6.2 One-way ANOVA . . . 61
6.3 ANOVA as a linear model* . . . 62
6.4 Means separation* . . . 64
6.5 One-way ANOVA from scratch* . . . 65
6.6 Answers . . . 66

7 Multivariate correlation and regression **68**
7.1 Multiple Correlation Analysis . . . 68
7.1.1 Pairwise simple correlations . . . 68
7.1.2 Pairwise partial correlations . . . 69
7.2 Multiple Regression Analysis . . . 72
7.3 Comparing regression models . . . 74

7.3.1 Comparing regression models with the adjusted R^2 74
7.3.2 Comparing regression models with the AIC 75
7.3.3 Comparing regression models with ANOVA 75
7.4 Stepwise multiple regression* 77
7.5 Combining discrete and continuous predictors 79
7.6 Diagnosing multi-colinearity 83
7.7 Visualising parallel regression* 87
7.8 Interactions* 88
7.9 Analysis of covariance* 92
7.10 Design matrices for combined models* 94
7.11 Answers 96

8 Factor analysis **99**
8.1 Principal components analysis 99
8.1.1 The synthetic variables* 101
8.1.2 Residuals* 103
8.1.3 Biplots* 109
8.1.4 Screeplots* 112
8.2 Factor analysis* 114
8.3 Answers 117

9 Geostatistics **119**
9.1 Postplots 119
9.2 Trend surfaces 119
9.3 Higher-order trend surfaces 125
9.4 Local spatial dependence and Ordinary Kriging 125
9.4.1 Spatially-explicit objects 129
9.4.2 Analysis of local spatial structure 132
9.4.3 Interpolation by Ordinary Kriging 133
9.5 Answers 138

10 Going further **140**

References **141**

Index of R concepts **146**

A Derivation of the hat matrix **146**
A.1 Influence of values on prediction 147

1 Introduction

This tutorial presents a data analysis sequence which may be applied to environmental datasets, using a small but typical data set of multivariate point observations. It is aimed at students in geo-information application fields who have some experience with basic statistics, but not necessarily with statistical computing. Five aspects are emphasised:

1. Placing statistical analysis in the framework of research questions;
2. Moving from simple to complex methods: first exploration, then selection of promising modelling approaches;
3. Visualising as well as computing;
4. Making correct inferences;
5. Statistical computation and visualization.

The analysis is carried out in the R environment for statistical computing and visualisation [16], which is an open-source dialect of the S statistical computing language. It is free, runs on most computing platforms, and contains contributions from top computational statisticians. If you are unfamiliar with R, see the monograph "Introduction to the R Project for Statistical Computing for use at ITC" [30], the R Project's introduction to R [28], or one of the many tutorials available via the R web page[1].

On-line help is available for all R methods using the `?method` syntax at the command prompt; for example `?lm` opens a window with help for the `lm` (fit linear models) method.

> **Note:** These notes use R rather than one of the many commercial statistics programs because R is a complete *statistical computing environment*, based on a modern computing language (accessible to the user), and with packages contributed by leading computational statisticians. R allows unlimited flexibility and sophistication. "Press the button and fill in the box" is certainly faster – but as with Windows word processors, "what you see is *all* you get". With R it may be a bit harder at first to do simple things, but you are not limited. R is completely free, can be freely-distributed, runs on all desktop computing platforms, is regularly updated, is well-documented both by the developers and users, is the subject of several good statistical computing texts, and has an active user group.

An introductory textbook with similar intent to these notes, but with a wider set of examples, is by Dalgaard [7]. A more advanced text, with many interesting applications, is by Venables and Ripley [35]. Fox [12] is an extensive explanation of regression modelling; the companion Fox and Weisberg [14] shows how to use R for this, mostly with social sciences datasets.

This tutorial follows a data analysis problem typical of earth sciences, natural and water resources, and agriculture, proceeding from visualisation and exploration through univariate point estimation, bivariate correlation and regression analysis, multivariate factor analysis, analysis of variance, and finally some geostatistics.

[1] http://www.r-project.org/

In each section, there are some *tasks*, for which a possible solution is shown as some `R code` to be typed at the console (or cut-and-pasted from the PDF version of this document, or loaded from the accompanying `.R` R code files). Then there are some *questions* to answer, based on the output of the task. Sample *answers* are found at the end of each section.

Optional sections

Some readers may want to skip more advanced sections or those that explain the mathematics behind the methods in more detail; these are marked with an asterisk '*' in the section title and in the table of contents.

Going further

These notes only scratch the surface of R's capabilities. In particular, the reader is encouraged to consult the on-line help as necessary to understand all the options of the methods used. Neither do these notes pretend to teach statistical inference; the reader should refer to a statistics reference as necessary; some good choices, depending on your background and the application, are Brownlee [3], Bulmer [4], Dalgaard [7] (general); Davis [9] (geology),Wilks [39] (meteorology); Snedecor and Cochran [31], Steel et al. [34] (agriculture); Legendre and Legendre [17] (ecology); and Webster and Oliver [38] (soil science).

See also §10, "Going further", at the end of the tutorial.

2 Example Data Set

This data set, fully described in Yemefack [40] and summarized in Yemefack et al. [41], contains 147 soil profile observations from the research area of the Tropenbos Cameroon Programme (TCP), representative of the humid forest region of southwestern Cameroon and adjacent areas of Equatorial Guinea and Gabon.

Three fixed soil layers (0–10 cm, 10–20 cm, and 30–50 cm) were sampled. The data set is from two sources. First, 45 representative soil profiles were described and sampled by genetic horizon. Soil characteristics for each of the three fixed layers were computed as weighted averages using genetic horizon thickness. Second, 102 plots from various land use/land cover types were sampled at the three fixed depths. Each of these samples was a bulked composite of five sub-samples taken with an auger in a plot diagonal basis. For both data sets, samples were located purposively and subjectively to represent soil and land use types. Laboratory analysis was by standard local methods [23].

For this exercise, we have selected three soil properties:

1. Clay content (code `Clay`), weight % of the mineral fine earth (< 2 mm);
2. Cation exchange capacity (code `CEC`), $cmol^{+}$ $(kg\ soil)^{-1}$
3. Organic carbon (code `OC`), volume % of the fine earth.

These three variables are related; in particular we know from theory and many detailed studies that the CEC of a soil depends on reactive sites, either on clay colloids or on organic complexes such as humus, where cations (such as K^{+} and Ca^{++}) can be easily adsorbed and desorbed [22, 32].

> The CEC is important for soil management, since it controls how much added artificial or natural fertiliser or liming materials will be

retained by the soil for a long-lasting effect on crop growth. Heavy doses of fertiliser on soils with low CEC will be wasted, since the extra nutrients will leach.

In addition, for each observation the following site information was recorded:

- East and North Coordinates, UTM Zone 32N, WGS84 datum, in meters (codes `e` and `n`)
- Elevation in meters above sea level (code `elev`)
- Agro-ecological zone, arbitrary code (code `zone`)
- Reference soil group, arbitrary code (code `wrb1`)
- Land cover type (code `LC`)

The soil group codes refer to Reference Groups of the World Reference Base for Soil Resources (WRB) , the international soil classification system [11]. These are presented in the text file as integer codes which correspond to three of the 31 Reference Groups identified worldwide, and which differ substantially in their properties and response to management [10]:

1. Acrisols (from the Haplic, Ferralic, and Plinthic subgroups)
2. Cambisols (from the Ferralic subgroup)
3. Ferralsols (from the Acri-ferric and Xanthic subgroups)

2.1 Loading the dataset

Note: The code in these exercises was tested with Sweave [18, 19] on R version 3.3.2 (2016-10-31), `sp` package Version: 1.2-4, `gstat` package Version: 1.1-5, and `lattice` package Version: 0.20-35 running on Mac OS X 10.6.3. So, the text and graphical output you see here was automatically generated and incorporated into LaTeX by running actual code through R and its packages. Then the LaTeX document was compiled into the PDF version you are now reading. Your output may be slightly different on different versions and on different platforms.

The dataset was originally prepared in a spreadsheet and exported as a text "comma-separated value" (CSV) file named `obs.csv`. This is a typical spreadsheet product with several inadequacies for processing in R, which we will fix up as we go along. This a tedious but necessary step for almost every dataset; so the techniques shown here should be useful in your own projects.

Task 1 : Start the R program and switch to the directory where the dataset is stored. •

Task 2 : Examine the contents of the CSV file. •

You can do this with a plain-text editor (*not* a spreadsheet) such as (in Windows) Notepad or Wordpad or (on Mac OS) TextEdit. We can also examine a file from within R, with the `file.show` method:

```
> file.show("obs.csv")

"e","n","elev","zone","wrb1","LC","Clay1","Clay2","Clay5","CEC1","CEC2","CEC5","OC1","
"1",702638,326959, 657,"2","3","FF",72,74,78,13.6,10.1, 7.1, 5.500,3.100,1.500
"2",701659,326772, 628,"2","3","FF",71,75,80,12.6, 8.2, 7.4, 3.200,1.700,1.000
"3",703488,322133, 840,"1","3","FV",61,59,66,21.7,10.2, 6.6, 6.980,2.400,1.300
...
"146",686534,339916, 445,"3","3","CF",34,40,45,13.2,12.2,11.7, 3.600,2.000,1.000
"147",688608,339579, 435,"3","3","BF",30,38,46, 6.9, 4.7, 2.9, 2.700,1.600,0.750
```

Q1 : *What is the format of the first line? What does it represent?* *Jump to A1* •

Q2 : *What is the format of the following lines? What do they represent?* *Jump to A2* •

Task 3 : Load the dataset into R using the `read.csv` method[2] and examine its structure. Identify each variable from the list above. Note its data type and (if applicable) numerical precision. •

```
> obs <- read.csv("obs.csv")

> str(obs)

'data.frame':        147 obs. of  15 variables:
 $ e    : int  702638 701659 703488 703421 703358 702334 681328 681508 681230 683989
 $ n    : int  326959 326772 322133 322508 322846 324551 311602 311295 311053 311685
 $ elev : int  657 628 840 707 670 780 720 657 600 720 ...
 $ zone : int  2 2 1 1 2 1 1 2 2 1 ...
 $ wrb1 : int  3 3 3 3 3 3 3 3 3 3 ...
 $ LC   : Factor w/ 8 levels "BF","CF","FF",..: 3 3 4 4 4 4 3 3 4 4 ...
 $ Clay1: int  72 71 61 55 47 49 63 59 46 62 ...
 $ Clay2: int  74 75 59 62 56 53 66 66 56 63 ...
 $ Clay5: int  78 80 66 61 53 57 70 72 70 62 ...
 $ CEC1 : num  13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6 7.9 14.9 ...
 $ CEC2 : num  10.1 8.2 10.2 8.4 9.2 11.6 7.4 7.1 5.7 6.8 ...
 $ CEC5 : num  7.1 7.4 6.6 8 8.5 6.2 5.4 7 4.5 6 ...
 $ OC1  : num  5.5 3.2 6.98 3.19 4.4 5.31 4.55 4.5 2.3 7.34 ...
 $ OC2  : num  3.1 1.7 2.4 1.5 1.2 3.2 2.15 1.42 1.36 2.54 ...
 $ OC5  : num  1.5 1 1.3 1.26 0.8 ...

> row.names(obs)

  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"
 [11] "11"  "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"
 [21] "21"  "22"  "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"
 [31] "31"  "32"  "33"  "34"  "35"  "36"  "37"  "38"  "39"  "40"
 [41] "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"  "49"  "50"
 [51] "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
```

[2] a wrapper for the very general `read.table` method

```
  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"
  [71] "71"  "72"  "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"
  [81] "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"
  [91] "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"  "100"
 [101] "101" "102" "103" "104" "105" "106" "107" "108" "109" "110"
 [111] "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
 [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130"
 [131] "131" "132" "133" "134" "135" "136" "137" "138" "139" "140"
 [141] "141" "142" "143" "144" "145" "146" "147"
```

Each variable has a *name*, which the import method `read.csv` reads from the first line of the CSV file; by default the first field (here, the observation number) is used as the row name (which can be accessed with the `row.names` method) and is not listed as a variable. The suffixes `1`, `2`, and `5` on the variable name roots `Clay`, `CEC`, and `OC` refer to the lower boundary of three depths, in dm; e.g. `OC5` is the organic C content of the 30–50 cm (3–5 dm) layer.

Each variable also has a *data type*. The import method attempts to infer the data type from the format of the data. In this case it correctly found that `LC` is a *factor*, i.e. has fixed set of codes. But it identified `zone` and `wrb1` as integers, when in fact these are coded factors. That is, the 'numbers' 1, 2, ... are just codes. R should be informed of their correct data type, which is important in linear models (§5.5) and analysis of variance (§6). In the case of the soils, we can also change the uninformative integers to more meaningful abbrevations, namely the first letter of the Reference Group name:

```
> obs$zone <- as.factor(obs$zone)
> obs$wrb1 <- factor(obs$wrb1, labels=c("a", "c", "f"))
```

Q3 : *What are the names, data types and numerical precision of the clay contents at the three depths?* *Jump to A3* •

Q4 : *What are the names, data types and numerical precision of the cation exchange capacities at the three depths?* *Jump to A4* •

You can save this as an R data object, so it can be read directly by R (not imported) with the `load` method; this will preserve the corrected data types.

```
> save(obs, file="obs.RData")
```

You can recover this dataset in another R session with the command:

```
> load(file="obs.RData")
```

2.2 A normalized database structure*

If you are familiar with relational database theory, the structure of our dataset may have bothered you, because it mixes the sample depth with the variable. For example, there are three fields for clay content (`Clay1`, `Clay2`, and `Clay5`), and similarly for organic C and CEC. How could we plot, for example, clay against

CEC for all the horizons together? There are several shortcuts but the most general solution is to change the database structure into a *normalized* set of relational tables:

1. The **observation points**, with a *primary key* that uniquely identifies the observation , with attributes that apply to the whole observation, namely:
   (a) the coordinates `e` and `n`
   (b) the elevation `elev`
   (c) the agro-ecological zone `zone`
   (d) the soil group `wrb1`
   (e) the land cover class `LC`
2. The **layers**, with a *primary key* made up of the primary key from the first table and the layer identification (1, 2, or 5), with attributes that apply to the horizon, namely:
   (a) `Clay`
   (b) `CEC`
   (c) `OC`

   Note that the first field of this primary key is also the *foreign key* into the first table.

For convenience we will also keep the original database structure for many of the analyses in this note.

There are several ways to do this in R; we will use the very flexible `reshape` method.

However, we first need to assign an observation ID to each record in the original table to use as the primary key. Here we can just use the row number:

```
> plot.id <- 1:dim(obs)[1]
```

Now we make the first table from those attributes of the observations that do not vary with layer:

```
> t.obs <- cbind(plot.id, obs[, 1:6])
> str(t.obs)

'data.frame':        147 obs. of  7 variables:
 $ plot.id: int  1 2 3 4 5 6 7 8 9 10 ...
 $ e      : int  702638 701659 703488 703421 703358 702334 681328 681508 681230 68398
 $ n      : int  326959 326772 322133 322508 322846 324551 311602 311295 311053 31168
 $ elev   : int  657 628 840 707 670 780 720 657 600 720 ...
 $ zone   : Factor w/ 4 levels "1","2","3","4": 2 2 1 1 2 1 1 2 2 1 ...
 $ wrb1   : Factor w/ 3 levels "a","c","f": 3 3 3 3 3 3 3 3 3 3 ...
 $ LC     : Factor w/ 8 levels "BF","CF","FF",..: 3 3 4 4 4 4 3 3 4 4 ...
```

Now we reshape the remainder of the fields into "long" format, beginning the plot ID (which is repeated three times, once for each layer), and dropping the fields that do not apply to the layers:

```
> t.layers <- cbind(plot.id = rep(plot.id, 3),
+                reshape(obs, direction="long",
+                drop=c("e", "n", "elev", "zone", "wrb1", "LC"),
+                varying=list(c("Clay1","Clay2","Clay5"),
+                c("CEC1", "CEC2", "CEC5"), c("OC1", "OC2", "OC5")),
+                times=c("1", "2", "5")))
> names(t.layers)[2:5] <- c("layer", "Clay", "CEC", "OC")
> t.layers$layer <- as.factor(t.layers$layer)
> str(t.layers)

'data.frame':        441 obs. of  6 variables:
 $ plot.id: int  1 2 3 4 5 6 7 8 9 10 ...
 $ layer  : Factor w/ 3 levels "1","2","5": 1 1 1 1 1 1 1 1 1 ...
 $ Clay   : int  72 71 61 55 47 49 63 59 46 62 ...
 $ CEC    : num  13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6 7.9 14.9 ...
 $ OC     : num  5.5 3.2 6.98 3.19 4.4 5.31 4.55 4.5 2.3 7.34 ...
 $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
```

The `reshape` method automatically created a new field `id` to uniquely identify the sample in the "long" format; there are 441 of these. It also created a field `times` to identify the vector from which each sample originated; this name due to `reshape`'s primary use with time series data. We renamed this field `layer`.

We now have a relational database structure, from which we can build temporary dataframes for a variety of queries.

Finally, we remove the temporary variable, and save the normalized data to a file as an R object:

```
> rm(plot.id)
> save(t.obs, t.layers, file="t.RData")
```

Answers

A1 : *The first line is a list of quoted field (variable) names, separated by commas. For example, the first field is named* `"e"`. *There are 15 field names.* *Return to Q1* •

A2 : *The other lines are the observations* (1 . . . 147)*; each observation is a list of values, one per field. There are 16 fields; the first is the observation ID, which has no name on the first line.* *Return to Q2* •

A3 : *The clay contents are* `Clay1`, `Clay2`, *and* `Clay5`*; these are integers (type* `int`*); their precision is 1%, i.e. they are specified to the nearest integer percent.* *Return to Q3* •

A4 : *The cation exchange capacities are* `CEC1`, `CEC2`, *and* `CEC5`*; these are floating-point numbers (type* `num`*); their precision is 0.1 cmol$^+$ (kg soil)$^{-1}$.* *Return to Q4* •

3 Research questions

A statistical analysis may be *descriptive*, simply reporting, visualizing and summarizing a data set, but usually it is also *inferential*; that is, statistical procedures are used as evidence to answer *research questions*. The most important of these are generally formulated by the researcher before data collection; indeed the sampling plan (number, location, strata, physical size) and data items should be motivated by the research questions. Of course, during field work or analysis other questions may suggest themselves from the data.

The data set for this case study was intended to answer at least the following research questions:

1. What are the *values* of soil properties important for agricultural production and soil ecology in the study area? In particular, the organic matter content (OM), proportion of clay vs. sand and silt (Clay), and the cation exchange capacity (CEC) in the upper 50 cm of the soil.[3]
    - OM promotes good soil structure, easy tillage, rapid infiltration and reduced runoff (hence less soil loss by surface water erosion); it also adsorbs nutrient cations and is a direct source of Nitrogen;
    - The proportion of clay has a major influence on soil structure, hardness, infiltration vs. runoff; almost all the nutrient cations not adsorbed on the OM are exchanged via the clay;
    - CEC is a direct measure of how well the soil can adsorb added cations from burned ash, natural animal and green manures, and artificial fertilizers.
2. What is the *inter-relation* (association, correlation) between these three variables? How much *total information* do they provide?
3. How well can CEC be *predicted* by OM, Clay, or both?
4. What is the depth profile of these variables? Are they constant over the first 50 cm depth; if not, how do they vary with depth?
5. Four agro-ecological zones and three major soil groups have been identified by previous mapping. Do the soil properties differ among these? If so, how much? Can the zones or soils groups be grouped or are they all different?
6. Each observation is located geographically. Is there a *trend* in any of the properties across the region? If so, how much variation does it explain, in which direction is it, and how rapidly does the property vary with distance?
7. Before or after taking any trend into account, is there any *local spatial dependence* in any of the variables?

These statistical question can then be used with knowledge of processes and causes to answer another set of research questions, more closely related to practical concerns or scientific knowledge:

[3] Note that the original data set included many more soil properties.

8. Is it necessary to do the (expensive) lab. procedure for CEC, or can it be predicted satisfactorily from the cheaper determinations for Clay and OM (or just one of these)?
9. Is it necessary to sample at depth, or can the values at depth be calculated from the values in the surface layer? If so, the cost of soil sampling could be greatly reduced.
10. Are the agro-ecological zones and/or soil maps a useful basis for predicting soil behaviour, and therefore a useful stratification for recommendations?
11. What soil-forming factor explains any regional trend?
12. What soil-forming factor explains any local spatial dependence?

Finally, the statistical questions can be used to *predict*:

13. How well can CEC be *predicted* by OM, Clay, or both?
14. What are the *expected values* of the soil properties, and the *uncertainties* of these predictions, at *unvisited locations* in the study area?

The last question can be answered by a predictive *map*.

4 Univariarte Analysis

Here we consider each variable separately.

4.1 Univariarte Exploratory Data Analysis

Task 4 : Summarise the clay contents at the three depths. •

To save typing, we first `attach` the `obs` data frame; this makes the field names in the data frame visible in the outer R environment; e.g. when we type `Clay1`, this field of the attached frame is accessed; otherwise we would have had to type `obs$Clay1`.

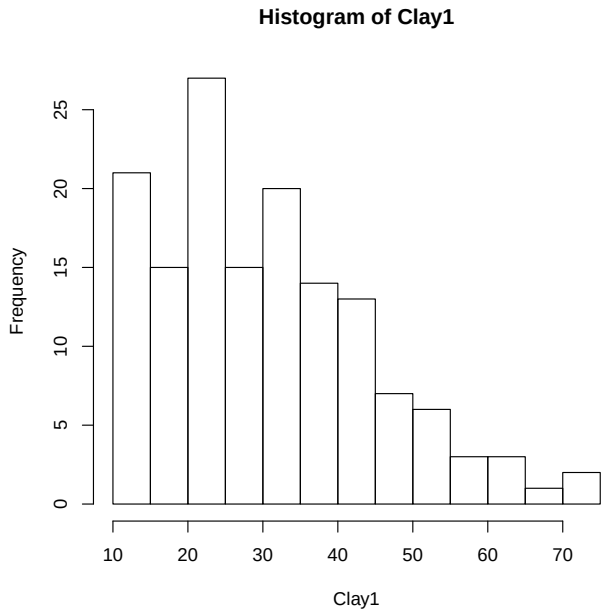
```
> attach(obs)
> summary(Clay1); summary(Clay2); summary(Clay5)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   10.0    21.0    30.0    31.3    39.0    72.0

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    8.0    27.0    36.0    36.7    47.0    75.0

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   16.0    36.5    44.0    44.7    54.0    80.0
```

Q5 : *What does the summary say about the trend of clay content with depth?* *Jump to A5* •

Q6 : *What evidence does the summary give that the distribution is somewhat symmetric?* *Jump to A6* •

Task 5 : Visualise the distribution of the topsoil clay content with a stem-and-leaf plot and a histogram. •

```
> stem(Clay1); hist(Clay1)

  The decimal point is 1 digit(s) to the right of the |

  1 | 000222233333444
  1 | 55555567788889999
  2 | 000011112222233344444
  2 | 555555555566788999
  3 | 00000001122222333333334444
  3 | 556666677889999
  4 | 022233334
  4 | 55555667899
  5 | 02334
  5 | 55689
  6 | 123
  6 | 7
  7 | 12

> hist(Clay1)
```

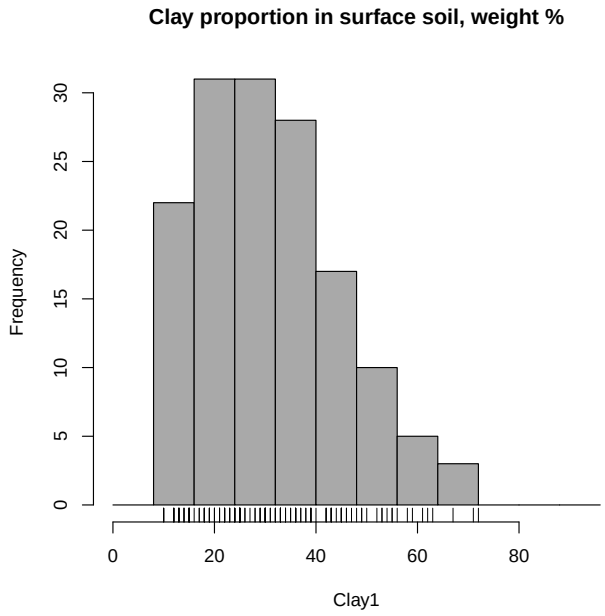


Q7 : *Does the distribution look symmetric? skewed? peaked?* *Jump to A7* •

It's easy to produce a much nicer and informative histogram:

All R graphics, including histograms, can be enhanced. Here we change the break points with the `breaks` argument, the colour of the bars with the `col` argument, the colour of the border with the `border` argument, and supply a title with the `main` argument; we then add a **rug** plot (with, what else?, the `rug` method) along the x-axis to show the actual observations.

```
> hist(Clay1, breaks=seq(0, 96, by=8), col="darkgray", border="black",
+     main="Clay proportion in surface soil, weight %")
> rug(Clay1)
```



Note the use of the `seq` ("sequence") method to make a list of break points. The `main=` argument is used to specify the main title; there is also a `sub=` argument for a subtitle.

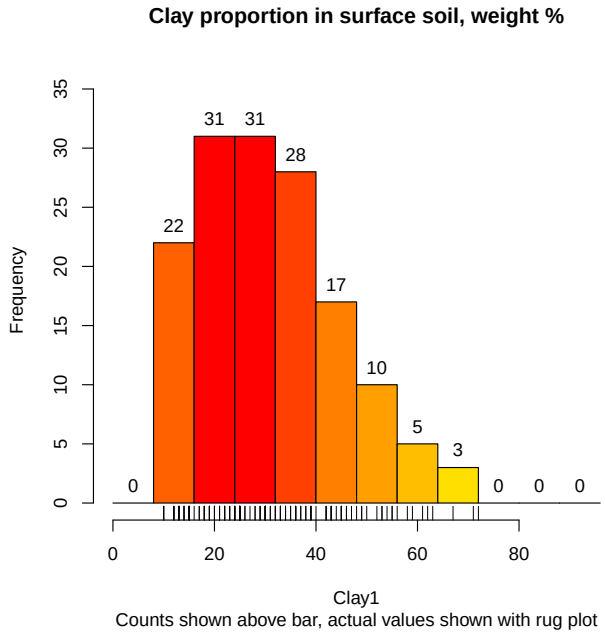
> **Note:** To see the list of named colours, use the `colors` command with no argument: `colors()`. There are many other ways to specify colours; see Rossiter [30, §5.5] and `?colors`.

Finally, we display a histogram with the actual counts. We first compute the histogram but don't plot it (`plot=F` argument), then draw it with the `plot` command, specifying a colour ramp, which uses the computed counts, and a title. Then the `text` command adds text to the plot at `(x, y)` positions computed from the class mid-points and counts; the `pos=3` argument puts the text on top of the bar.

```
> h <- hist(Clay1, breaks=seq(0, 96, by=8), plot=F)
> str(h)

List of 6
 $ breaks  : num [1:13] 0 8 16 24 32 40 48 56 64 72 ...
 $ counts  : int [1:12] 0 22 31 31 28 17 10 5 3 0 ...
 $ density : num [1:12] 0 0.0187 0.0264 0.0264 0.0238 ...
 $ mids    : num [1:12] 4 12 20 28 36 44 52 60 68 76 ...
```

```
 $ xname   : chr "Clay1"
 $ equidist: logi TRUE
 - attr(*, "class")= chr "histogram"
```

```
> plot(h, col = heat.colors(length(h$mids))[length(h$count)-rank(h$count)+1],
+    ylim = c(0, max(h$count)+5),
+    main="Clay proportion in surface soil, weight %",
+    sub="Counts shown above bar, actual values shown with rug plot")
> rug(Clay1)
> text(h$mids, h$count, h$count, pos=3)
> rm(h)
```



We can see that there are a few unusually high values. The record for these should be examined to see if there is anything unusual about it.

Task 6 : Display the entire record for observations with clay content of the topsoil over 65%. •

There are (at least) two ways to do this. First we show the easy way, with a condition to select rows:

```
>     obs[Clay1 > 65, ]

          e      n elev zone wrb1 LC Clay1 Clay2 Clay5 CEC1 CEC2 CEC5
1    702638 326959  657    2    f FF    72    74    78 13.6 10.1  7.1
2    701659 326772  628    2    f FF    71    75    80 12.6  8.2  7.4
106 696707 327780  623    2    f FV    67    70    73 22.0 13.0 11.0
    OC1 OC2 OC5
1   5.5 3.1 1.5
2   3.2 1.7 1.0
106 4.8 2.1 1.2
```

We can get the same effect by identifying the rows and then using these as row

indices:

```
>     (ix <- which(Clay1 > 65)); obs[ix, ]

[1]   1   2 106

         e      n elev zone wrb1 LC Clay1 Clay2 Clay5 CEC1 CEC2 CEC5
1   702638 326959  657    2    f FF    72    74    78 13.6 10.1  7.1
2   701659 326772  628    2    f FF    71    75    80 12.6  8.2  7.4
106 696707 327780  623    2    f FV    67    70    73 22.0 13.0 11.0
    OC1 OC2 OC5
1   5.5 3.1 1.5
2   3.2 1.7 1.0
106 4.8 2.1 1.2
```

Q8 : *Which are the unusual observations? Is there any evidence of errors in data entry? Why or why not?* *Jump to A8* •

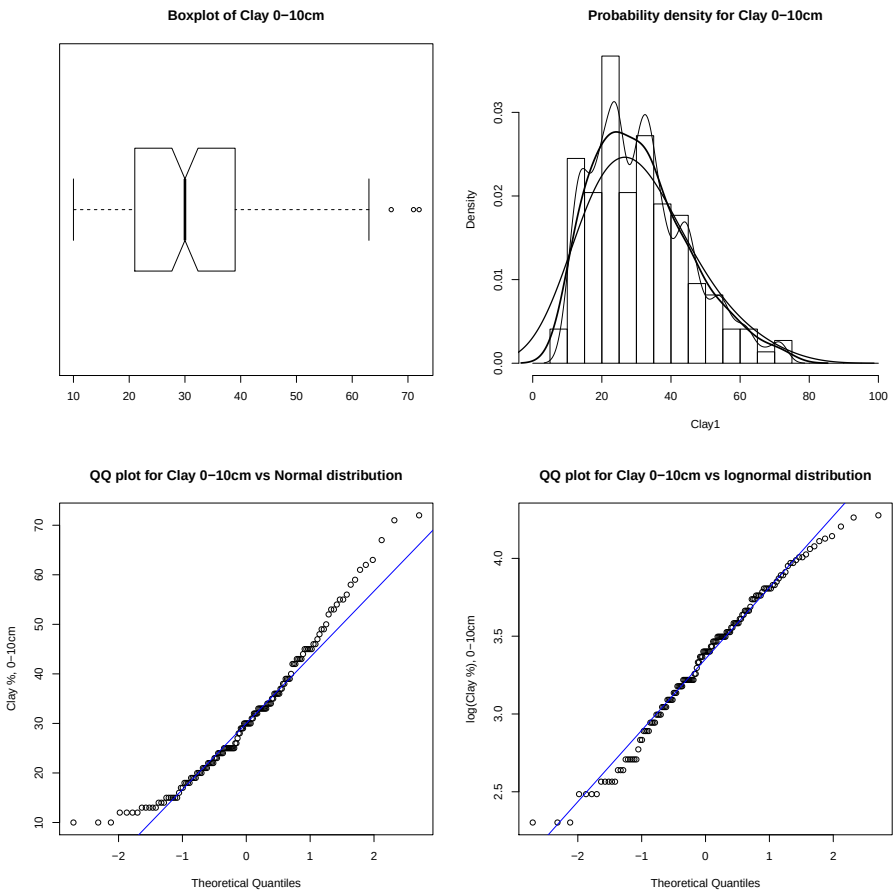
Other exploratory graphics There are several other ways to view the distribution of a single variable in a histogram:

1. We can specify the number of histogram bins and their limits.
2. We can view the histogram as a *probability density* rather than a frequency (actual number of cases); this makes it easier to compare histograms with different numbers of observations.
3. We can compare the actual distribution of a variable to a theoretical distribution with a *quantile-quantile* plot.
4. We can fit empirical *kernel density estimator* curves, which give a more-or-less smoothed continuous approximation to the histogram.

Task 7 : Show the distribution as a boxplot. Plot the histogram with bins of 5% clay, with kernel density estimators. Make a quantile-quantile plot for both the normal and lognormal distributions. •

```
> par(mfrow=c(2,2))
>     boxplot(Clay1, notch=T, horizontal=T,
+       main="Boxplot of Clay 0-10cm")
> #
> hist(Clay1, freq=F, breaks=seq(0,100,5),
+      main="Probability density for Clay 0-10cm")
> lines(density(Clay1),lwd=2)
> lines(density(Clay1, adj=.5),lwd=1)
> lines(density(Clay1, adj=2),lwd=1.5)
> #
> qqnorm(Clay1, main="QQ plot for Clay 0-10cm vs Normal distribution",
+    ylab="Clay %, 0-10cm")
> qqline(Clay1, col=4)
> #
> qqnorm(log(Clay1), main="QQ plot for Clay 0-10cm vs lognormal distribution",
+    ylab="log(Clay %), 0-10cm")
```

```
> qqline(log(Clay1), col=4)
> par(mfrow=c(1,1))
```



The boxplot (upper-left) matches the histogram: the distribution is right-skewed. The three largest observations are shown as *boxplot outliers*, i.e. they are more than 1.5 times the inter-quartile range (width of the box) larger than the 3rd quartile. This is just a technical measure: they are *boxplot* outliers, but this does not necessarily mean that they are part of a different population. In particular, a few boxplot outliers are expected in the direction of skew.

Q9 : *Does the distribution look normal or lognormal? What does this imply for the underlying natural process?* *Jump to A9* •

Exercise 1 : Repeat the analysis with the clay content of the 10–20 cm or 30–50 cm layers; comment on any differences with the distribution in the 0–10 cm layer.

4.2 Point estimation; inference of the mean

When computing summary statistics (§4.1), we calculated a *sample mean*; this is simply a descriptive statistic for that sample. If we go one step further, we can

ask what is the best estimate of the *population* mean, given this sample from that population; this is an example of *point estimation*. We may also make *inferences* about the true (but unknown) mean of the population: is it equal to a known value, or perhaps higher or lower?

For small samples, inference must be based on the t-distribution. The *null hypothesis* can be a value known from theory, literature, or previous studies.

Task 8 : Compute the best estimate of the population mean of topsoil clay content from this sample, its 99% confidence interval, and the probability that it is not equal to 30% clay. •

```
> t.test(Clay1, mu=30, conf.level=.99)

        One Sample t-test

data:  Clay1
t = 1.11, df = 146, p-value = 0.27
alternative hypothesis: true mean is not equal to 30
99 percent confidence interval:
 28.272 34.272
sample estimates:
mean of x
   31.272
```

Q10 : *What is the estimated population mean and its 99% confidence interval? Express this in plain language. What is the probability that we would commit a Type I error if we reject the null hypothesis that the population mean is 30% clay?* *Jump to A10* •

Sometimes we are interested in the mean with relation to a set *threshold* value; this usually comes from external considerations such a regulations or an existing classification system.

Q11 : *What is the probability that the true population mean is less than 35% clay? (Hint: use the* `alternative="less"` *argument to the* `t.test` *method.)* *Jump to A11* •

4.3 Answers

A5 : *It increases with depth, as evidenced by the mean, quartiles including the median, and maximum.* *Return to Q5* •

A6 : *The mean and median are almost equal.* *Return to Q6* •

A7 : *Both the stem-and-leaf plot and the histogram show that, compared to a normal*

distribution, this is skewed towards positive values and with a lower peak. *Return to Q7* •

A8 : *Observations 1, 2, and 106 in the list of 147 observations have surface clay contents over 65%. These seem consistent with the clay contents of the other layers, so there is no evidence of a data entry error.* *Return to Q8* •

A9 : *It is* not normal*; especially at the lower tail of the normal distribution where values are too high. This implies that clay content of the topsoil does not simply reflect an addition of many small errors.*

It is not lognormal*; especially at the upper tail of the lognormal distribution where values are too low. This implies that clay content of the topsoil does not simply reflect a multiplication of many small errors. So, there should be some underlying process, i.e. an identifiable cause of the variation across the sample set.* *Return to Q9* •

A10 : *The best estimate of the mean is* 31.3% *clay. With only 1% chance of being wrong, we assert that the true mean is between about* 28.3 *and* 34.3% *clay. We can not reject the null hypothesis; if we do, there is about a 0.27 probability (more than 1 in 4) that we are wrong.* *Return to Q10* •

A11 : p = 0.00073*, i.e. it is almost certain that the mean is below this threshold.* *Return to Q11* •

5 Bivariate correlation and regression

Now we consider the relation between two variables. This is the cause of much confusion and incorrect analysis.

5.1 Conceptual issues in correlation and regression

Correlation and various kinds of regression are often misused. There are several good journal articles that explain the situation, with examples from earth science applications [21, 36]. A particularly understandable introduction to the proper use of regression is by Webster [37], whose notation we will use.

Bivariate correlation and regression both compare two variables that refer to *the same observations*, that is, they are *paired*. This is the natural order in a data frame: each *row* represents *one observation* on which several *variables* were measured; in the present case, the coördinates, clay contents, organic matter contents, and CEC at three depths, so we can use the sample to ask about the relation between these variables in the whole population.

First we discuss the key issues; then we resume the analysis in §5.2.

Description vs. prediction, relation vs. causation Regression analysis can be used for two main purposes:

1. To *describe* a relation between two or more variables;

2. To *predict* the value of a variable (the *predictand*, sometimes called the *dependent* variable or *response*), based on one or more other variables (the *predictors*, sometimes called *independent* variables.

So the analyst must first decide whether the results of the analysis will be used predict or not. These can lead to different mathematical procedures.

Another pair of concepts which are sometimes confused with the above are related to the *philosophical issues of knowledge*:

1. The *relation* between two or more variables, often described mathematically as the *correlation* ('co-relation');
2. The *causation* of one variable by another.

This second pair is a much stronger distinction than the first. The issue of causation must also involve some *conceptual model* of how the two phenomena are related. **Statistics can never prove causation**; it can only provide evidence for the strength of a causative relation supported by other evidence.

Types of models A *simple* correlation or regression relates two variables only; a *multiple* correlation or regression relates several variables at the same time. Modelling and interpretations are much trickier in the multivariate case, because of the inter-relations between the variables.

A *linear* relation models one variable as a linear function of one or several other variables. That is, a proportional change in the predictor results in a proportional change in the predictand or the modelled variable. Any other relation is *non-linear*, but there is controversy over the use of this term. In particular, a *polynomial* model, where one variable is modelled as a sum of one or more *powers* of one or more other variables, is termed *curvilinear* and is usually considered a linear model.

Non-linear relations may be *linearisable* by means of a transformation of one or more variables, but in many interesting cases this is not possible; these are *intrinsically non-linear*.

Fixed vs. random variables* An important distinction is made between predictors which are known without error, whether fixed by the experimenter or measured, and those that are not. Webster [37] calls the first type a "Gauss linear model", because only the predictand has error, whereas the predictor is a *mathematical* variable, as opposed to a *random* variable which is measured with error. The regression goes in one direction only, from the mathematical predictor to the random response, and is modelled by a **linear model with error**:

$$y_i = BX_i + \varepsilon_i$$

of which the simplest case is a line with intercept:

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

Note that there is no error associated with the predictors x_i, only with the predictand y_i. Thus the predictors are assumed to be known without error, or at least the error is quite small in comparison to the error in the model. An example of this type is a designed agricultural experiment where the quantity of fertiliser added (the predictor) is specified by the design and the crop yield is measured (the predictand); there is random error ε_i in this response.

An example of the second type is where the crop yield is the predictand, but the predictor is the measured nutrient content of the soil. Here we are modelling the relation as a **bivariate normal distribution** of two random variables, X and Y with (unknown) population means μ_X and μ_Y, (unknown) population variances σ_X and σ_Y, and an (unknown) correlation ρ_{XY} which is computed as the standardised (unknown) covariance $\mathrm{Cov}(X, Y)$:

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

In practice, the distinction between the two models is not always clear. The predictor, even if specified by the experimenter, can also have some measurement error. In the fertiliser experiment, even though we specify the amount per plot, there is error in measuring, transporting, and spreading it. In that sense it can be considered a random variable. But, since we have some control over it, the experimental error can be limited by careful procedures. We can not limit the error in the response by the same techniques.

5.2 Bivariate Exploratory Data Analysis

The first question in the analysis is the relation between clay content in the three layers. We could have several specific questions:

1. Are the clay contents between layers positively, negatively, or not *related*? E.g. if the topsoil is high in clay, does that imply that the subsoil is high also? low? or that we can't tell.
2. How can we explain this relation? I.e., what does it imply for soil formation in this area?
3. How well can we *predict* the subsoil clay from the topsoil? If we can do this, it would save fieldwork (having to auger half a meter from the surface) and laboratory work (having to analyse another sample).
4. What is the predictive equation?

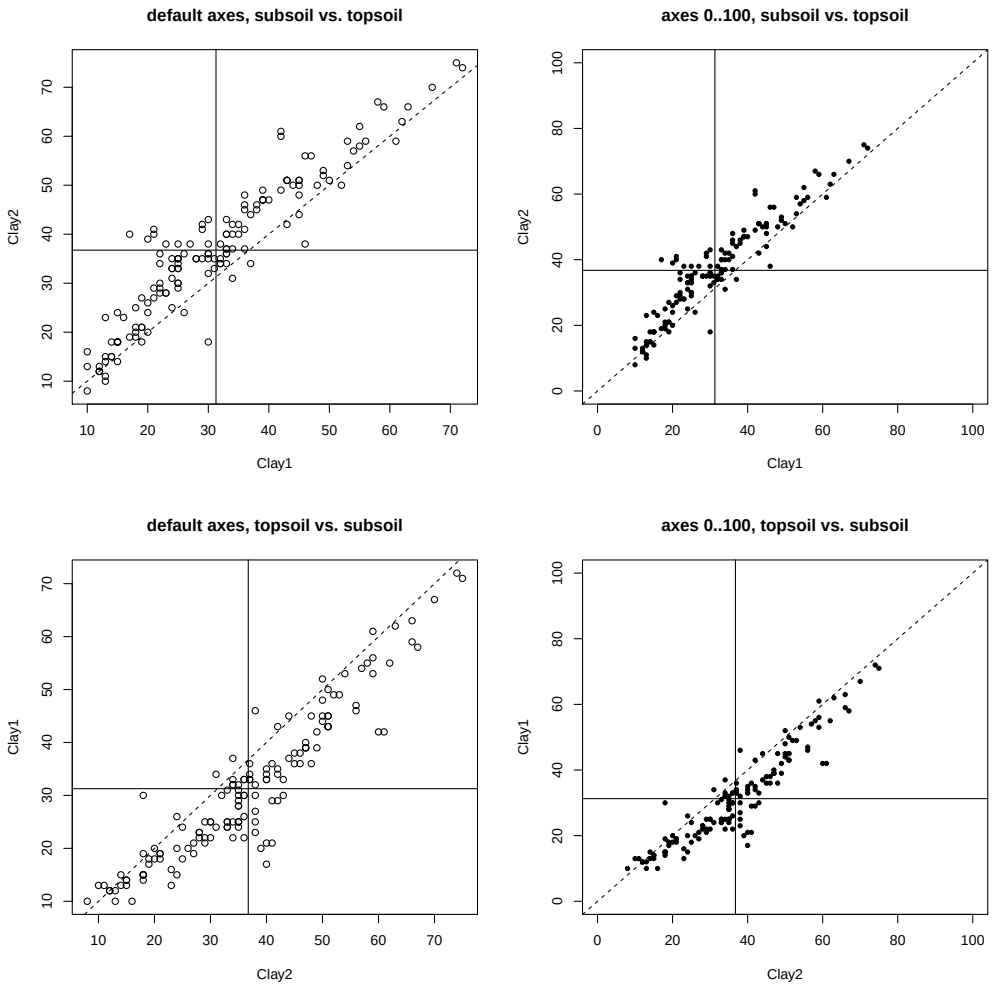
Note that the second question, requiring a conceptual model and support from other information, is much harder to answer than the first, requiring only a mathematical manipulation.

Task 9 : View the relation between layers. •

Here are two ways to show the same scatterplot; in the second we specify the plotting limits of each axis. We also show the inverted plots. In all graphs we show the 1:1 line and the means of both variables. We plot these all in the same figure, using the `mfrow` argument to the `par` ("graphics parameters") method on the open graphics device.

> **Note:** To see all the possible printing characters (the `pch=` argument to `plot`), run the command `example(plot.)`

```
> par(mfrow=c(2,2))   # 2x2 matrix of plots in one figure
> #
> plot(Clay1,Clay2); abline(0,1,lty=2);
> title("default axes, subsoil vs. topsoil")
> abline(h=mean(Clay2)); abline(v=mean(Clay1))
> #
> plot(Clay1,Clay2,xlim=c(0,100),ylim=c(0,100),pch=20); abline(0,1,lty=2)
> title("axes 0..100, subsoil vs. topsoil")
> abline(h=mean(Clay2)); abline(v=mean(Clay1))
> #
> plot(Clay2,Clay1); abline(0,1,lty=2)
> title("default axes, topsoil vs. subsoil")
> abline(h=mean(Clay1)); abline(v=mean(Clay2))
> #
> plot(Clay2,Clay1,xlim=c(0,100),ylim=c(0,100),pch=20); abline(0,1,lty=2)
> title("axes 0..100, topsoil vs. subsoil")
> abline(h=mean(Clay1)); abline(v=mean(Clay2))
> #
> par(mfrow=c(1,1))  # reset to 1 plot per figure
```



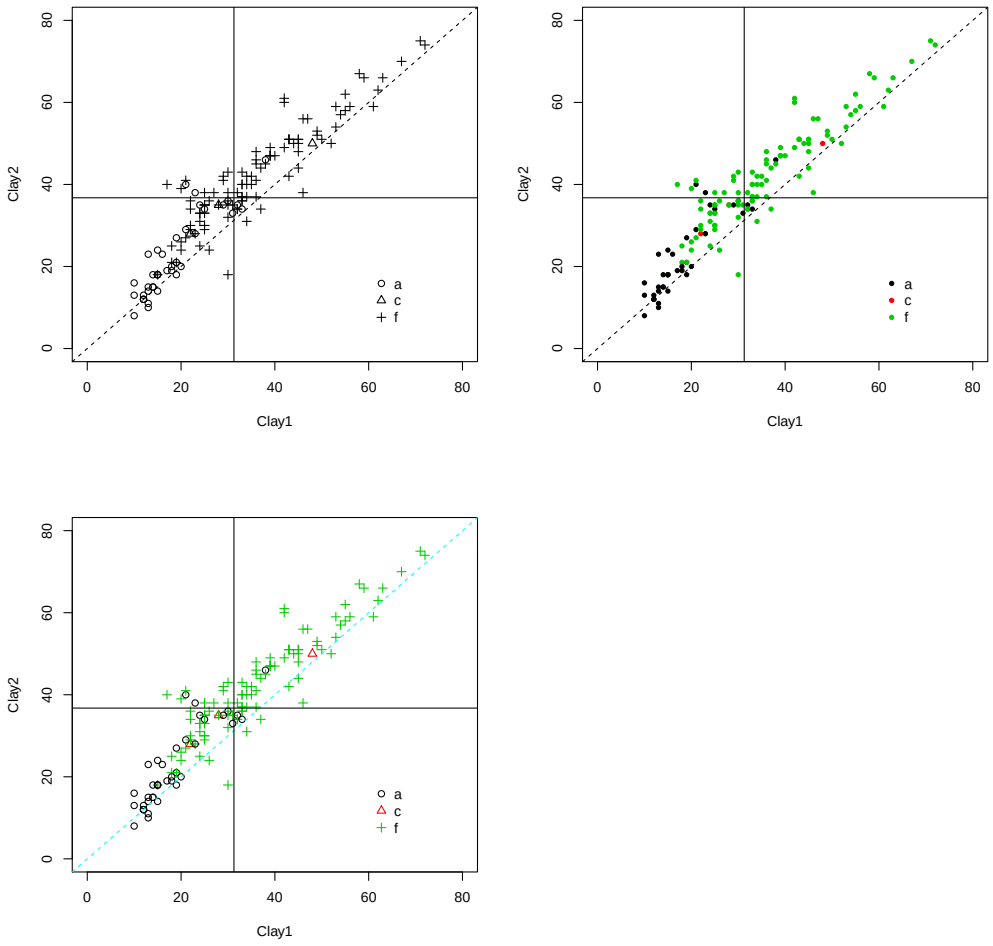
Q12 : *Describe the relation in words.* *Jump to A12* •

Task 10 : **Optional** but interesting: View the relation between layers, showing whether it is the same for each of the three soil classes (code `wrb1`). •

We can show the soil class either by symbol or by colour (or both); here we compare the three methods on one plot:

```
> par(mfrow=c(2,2))
> #
> plot(Clay1, Clay2, xlim=c(0,80), ylim=c(0,80),
+      pch=as.numeric(wrb1))
> abline(0,1,lty=2)
> abline(h=mean(Clay2)); abline(v=mean(Clay1))
> legend(60, 20, legend=levels(wrb1), pch=1:nlevels(wrb1), bty="n")
> #
> plot(Clay1, Clay2, xlim=c(0,80), ylim=c(0,80), pch=20,
+      col=as.numeric(wrb1))
```

```
> legend(60, 20, legend=levels(wrb1), pch=20,
+          col=1:nlevels(wrb1), bty="n")
> abline(0, 1, lty=2)
> abline(h=mean(Clay2)); abline(v=mean(Clay1))
> #
> plot(Clay1, Clay2, xlim=c(0,80), ylim=c(0,80),
+      pch=as.numeric(wrb1), col=as.numeric(wrb1))
> abline(0, 1, lty=2, col=5)
> abline(h=mean(Clay2)); abline(v=mean(Clay1))
> legend(60, 20, levels(wrb1), pch=1:nlevels(wrb1),
+      col=1:nlevels(wrb1), bty="n")
> #
> par(mfrow=c(1,1))
```



Note the use of the `levels` method to extract the soil codes for use in the legend, and the use of the `as.numeric` method to convert the soil code to an integer for use with the `col=` and `pch=` graphics parameters.

Q13 : *Is there any difference in the relation between soil classes?* *Jump to A13*

•

5.3 Bivariate Correlation Analysis

If the two variables to be correlated are numeric and relatively symmetric, we use the standard *Pearson's product-moment correlation.*

The *sample covariance* is computed as:

$$\text{Cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}\{(x_i-\bar{x})(y_i-\bar{y})\}$$

and then the *sample Pearson's correlation coefficient* is computed as:

$$r_{XY} = \text{Cov}(X,Y)/s_X \cdot s_Y$$

Task 11 : Compute the Pearson's correlation between the clay contents of the topsoil and subsoil. Test whether this correlation is significant. •

First we compute the correlation from the sample covariance (computed with the `cov` method) and standard deviations (computed with the `sd` method), to show how the definition works, then we use the `cor.test` method to compute a confidence interval.

```
> sum((Clay2-mean(Clay2))*(Clay1-mean(Clay1)))/(length(Clay2)-1)

[1] 190.74

> cov(Clay1,Clay2)

[1] 190.74

> sd(Clay1); sd(Clay2)

[1] 13.936

[1] 14.626

> cov(Clay1,Clay2)/(sd(Clay1)*sd(Clay2))

[1] 0.9358

> cor(Clay1,Clay2)

[1] 0.9358

> cor.test(Clay1,Clay2)

        Pearson's product-moment correlation

data:  Clay1 and Clay2
t = 32, df = 145, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.91208 0.95327
```

```
sample estimates:
   cor
0.9358
```

Q14 : *According to this test, what is the probability that the two clay contents, in the population from which this sample was taken, are in fact* not *correlated?* *Jump to A14* •

Q15 : *What is the best estimate for the population correlation coefficient? With only 5% probability of being wrong, what are the lowest and highest values this coefficient could in fact have?* *Jump to A15* •

5.4 Fitting a regression line

When we decide to consider one of the variables as as a response and the other as a predictor, we attempt to fit a line that best describes this relation. There are three types of lines we can fit, usually in this order:

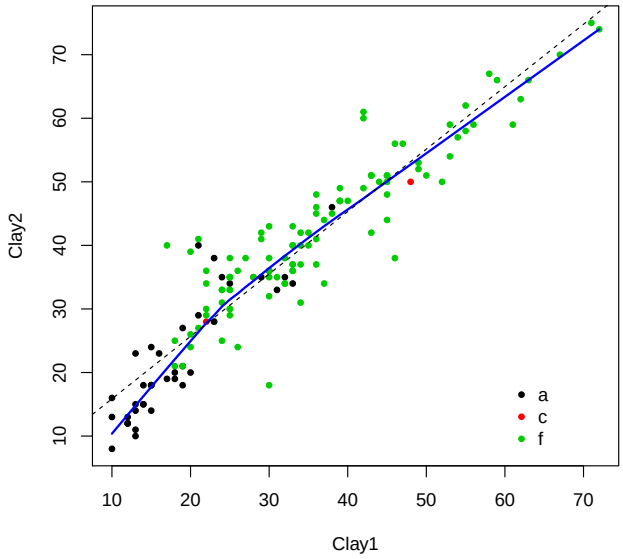
1. Exploratory, non-parametric
2. Parametric
3. Robust

The first kind just gives a "smooth" impression of the relation. The second fits according to some optimality criterion; the classic *least-squares* estimate is in this class. The third is also parametric but optimises some criterion that protects against a few unusual data values in favour of the majority of the data.

A common non-parametric fit is the LOWESS ("locally weighted regression and smoothing scatterplots") [35], computed by R method `lowess`. This has a user-adjustable parameter, the smoother's "span", which is the proportion of points in the plot which influence the smooth at each value; larger values result in a smoother plot. This allows us to visualise the relation either up close (low value of parameter) or more generally (high). The default is 2/3.

Task 12 : Plot subsoil vs. surface soil clay with the default smooth line. Show the soil type of each point by its colour. For comparison, plot the least-squares fit with a thin dashed line (n.b. this is not explained until 5.5). •

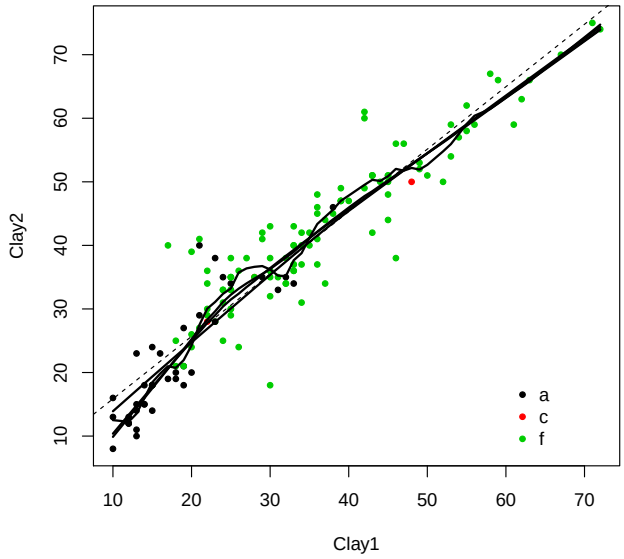
```
> plot(Clay2 ~ Clay1, pch=20,col=as.numeric(wrb1))
> legend(60, 20, legend=levels(wrb1), pch=20, col=1:nlevels(wrb1), bty="n")
> lines(lowess(Clay1, Clay2), lwd=2, col="blue")
> abline(lm(Clay2 ~ Clay1), lty=2)
```



Q16 : *What is the difference between the "best" line and the smooth fit? Does the smooth fit provide evidence that the different soil types might have different relations between subsoil and surface soil clay content?* *Jump to A16* •

Task 13 : Plot subsoil vs. surface soil clay with the default smooth line and with 1/10, 1/2, and all the points contributing to the fit. •

```
> plot(Clay1,Clay2,pch=20,col=as.numeric(wrb1))
> legend(60, 20, legend=levels(wrb1),pch=20, col=1:3,bty="n")
> for (f in c(0.1, .5, 2/3, 1)) {
+   lines(lowess(Clay1, Clay2, f=f), lwd=2) }
> abline(lm(Clay2 ~ Clay1), lty=2)
```



Q17 : *What happens as the smoothness parameter changes? Which value gives the best visualisation in this case?* *Jump to A17* •

5.5 Bivariate Linear Regression

Both subsoil and topsoil clay are measured with the same error, so the *bivariate normal model* is appropriate. That means we can compute the regression in both directions.

Subsoil as predicted by topsoil We may want to predict subsoil clay from topsoil clay. If we can establish this relation, we wouldn't have to sample the subsoil, just the topsoil; this is easier and also saves laboratory analysis.

Task 14 : Compute the ordinary least-squares (OLS) linear regression of subsoil clay on surface soil clay •

The `lm` method by default computes the OLS solution:

```
> lm21<-lm(Clay2 ~ Clay1)
> summary(lm21)

Call:
lm(formula = Clay2 ~ Clay1)

Residuals:
    Min      1Q  Median      3Q     Max
-17.499  -3.463   0.143   2.662  17.269

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)   6.0354     1.0514    5.74  5.3e-08 ***
Clay1         0.9821     0.0307   31.96  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.17 on 145 degrees of freedom
Multiple R-squared:  0.876,        Adjusted R-squared:  0.875
F-statistic: 1.02e+03 on 1 and 145 DF,  p-value: <2e-16
```

Q18 : *What is the best predictive equation for subsoil clay, given topsoil clay?* *Jump to A18* •

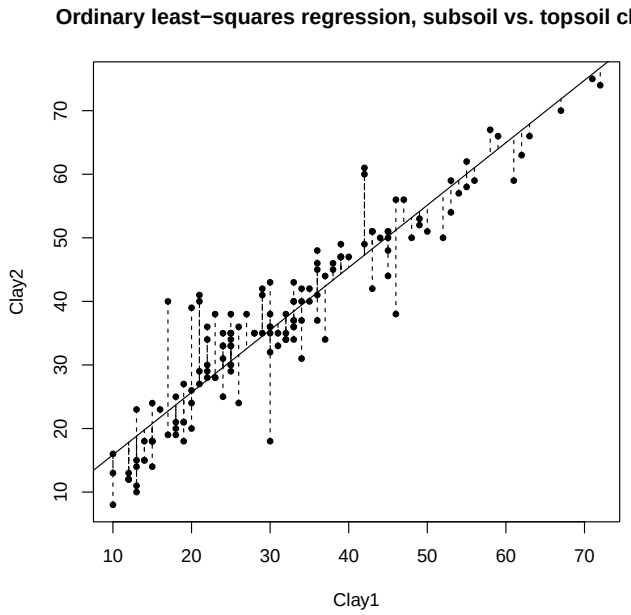
Q19 : *Express this in plain language: (1) How much subsoil clay is predicted for a soil with no topsoil clay? (2) How much does subsoil clay increase for a given increase in topsoil clay?* *Jump to A19* •

Q20 : *How much of the total variation in subsoil clay among the 147 samples is explained by topsoil clay?* *Jump to A20* •

Visualising the regression Here we show what the regression line looks like, and visualise the sense it which it is the "best" possible line.

Task 15 : Plot the least-squares regression line on the scatterplot of subsoil vs. topsoil clay, showing the residuals (distance to best-fit line). •

```
> plot(Clay1, Clay2, pch=20)
> title("Ordinary least-squares regression, subsoil vs. topsoil clay")
> abline(lm21)
> segments(Clay1, Clay2, Clay1, fitted(lm21), lty=2)
```



Q21 : *What would happen to the total length of the residual lines if the "best-fit" regression line were moved up or down (changed intercept) or if its slope were changed?* *Jump to A21* •

Reversing the regression: topsoil as predicted by subsoil As explained above, mathematically we can compute the regression in either direction.

Task 16 : Compute the regression of topsoil on subsoil clay •

This is the inverse of the previous regression.

```
> lm12<-lm(Clay1 ~ Clay2) ; summary(lm12)

Call:
lm(formula = Clay1 ~ Clay2)

Residuals:
    Min      1Q  Median      3Q     Max
-17.172  -2.534  -0.097   2.795  15.445

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.4949     1.1028   -1.36     0.18
Clay2         0.8917     0.0279   31.96   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.93 on 145 degrees of freedom
Multiple R-squared:  0.876,        Adjusted R-squared:  0.875
F-statistic: 1.02e+03 on 1 and 145 DF,  p-value: <2e-16
```

Q22 : *What is the best predictive equation for topsoil clay, given subsoil clay?* *Jump to A22* •

Q23 : *Express this in plain language: (1) How much topsoil clay is predicted for a soil with no subsoil clay? (2) How much does topsoil clay increase for a given increase in subsoil clay?* *Jump to A23* •

Q24 : *How much of the total variation in subsoil clay among the 147 samples is explained by topsoil clay?* *Jump to A24* •

Task 17 : Explain the differences between the two relations. •

Q25 : *Wait a minute! The first equation says that subsoil clay is about 6% higher than topsoil, while the second one says that topsoil is about 1.5% lower than the subsoil. Shouldn't these two equations give the same difference?* *Jump to A25* •

Q26 : *Wait a minute! The first equation says that the proportional increase in subsoil clay for 1% in topsoil is about 0.98%; the inverse of this is* $1/0.98 = 1.02$*, yet the second equation says the proportional increase in topsoil clay for 1% in subsoil is only 0.89%. Shouldn't these two equations give the same slope?* *Jump to A26* •

5.6 Bivariate Regression Analysis from scratch*

In this optional section we compute the regression coefficients directly from their definitions, rather than using the `lm()` function. This gives a better insight into the meaning of the coefficients. In practice, we would use the `lm()` function.

Task 18 : Compute the sample variance for both variables, and their sample covariance. These will be used to compute the regressions. •

```
> s2x<-var(Clay1); s2x

[1] 194.21

> s2y<-var(Clay2); s2y

[1] 213.92

> sxy<-var(Clay1,Clay2); sxy

[1] 190.74
```

Note that the variances are of similar magnitude. We can compute the variances and co-variances directly from their definitions, just to check that R is doing the right thing. This is a nice illustration of R's implicit vector arithmetic:

```
> sum((Clay1-mean(Clay1))^2)/(length(Clay1)-1)

[1] 194.21

> sum((Clay2-mean(Clay2))^2)/(length(Clay2)-1)

[1] 213.92

> sum((Clay2-mean(Clay2))*(Clay1-mean(Clay1)))/(length(Clay1)-1)

[1] 190.74
```

Task 19 : Compute the slopes and intercepts (y on x and x on y). •

For the regression of y on x, these are estimated by from the sample covariance and variances as:

$$\begin{aligned} \hat{\beta}_{Y.x} &= s_{XY}/s_x^2 \\ \hat{\alpha}_{Y.x} &= \bar{y} - \hat{\beta}_{Y.x}\bar{x} \end{aligned}$$

For the inverse regression, i.e. x on y, the estimates are:

$$\begin{aligned} \hat{\beta}_{X.y} &= s_{XY}/s_y^2 \\ \hat{\alpha}_{X.y} &= \bar{x} - \hat{\beta}_{X.y}\bar{y} \end{aligned}$$

Note that in both cases the regression line passes through the centroid estimated by the means, i.e. $(\bar{x}, \bar{y})$.

We compute these in R from the sample variances and covariances calculated above:

```
> byx<-sxy/s2x; byx

[1] 0.98212

> ayx<-mean(Clay2)-byx*mean(Clay1); ayx

[1] 6.0354

> bxy<-sxy/s2y; bxy

[1] 0.89166

> axy<-mean(Clay1)-bxy*mean(Clay2); axy

[1] -1.4949
```

These are the same coefficients we got from the `lm` method.

5.7 Regression diagnostics

The `lm` method will usually compute a fit, i.e. give us the *mathematical* answer to the question "What is the best linear model to explain the observations?". The model's adjusted R^2 tells us how well it fits the observations overall; this is the highest-possible R^2 with the given predictor.

However, the model may not be *statistically* adequate:

- The fit may not be equally-good over the whole range of observations, i.e. the error may not be independent of the predictor;
- The assumptions underlying least-squares regression may not be met, in particular, that the residuals are normally-distributed.

So for any regression, we should examine some *diagnostics* of its success and validity. Here we look at (1) the fit to observed data; (2) unusually-large residuals; (3) distribution of residuals; (4) points with unusual leverage.

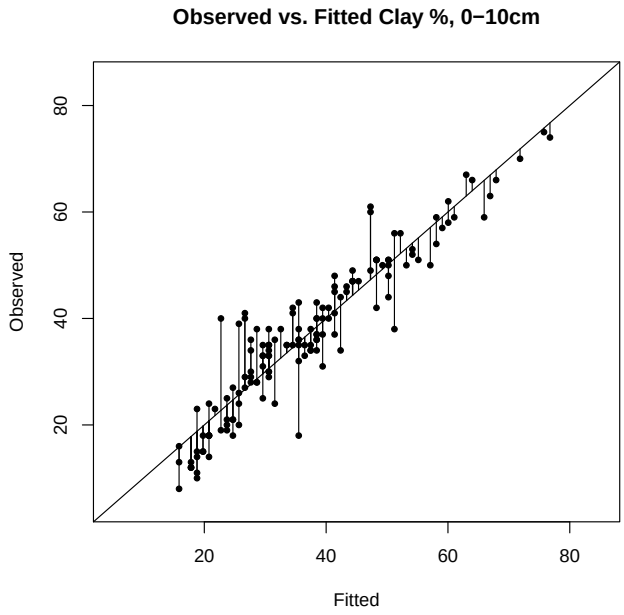
5.7.1 Fit to observed data

The first diagnostic is how well the model fits the data; this is the success of the model *conditional on the sample data*; this does not yet say how well the model is expected to fit the entire *population*.

The `fitted` method applied to a linear model object returns values predicted by a model at the observation values of the predictor. This is applied to an object saved from the `lm` method. Similarly, the `resid` method returns the *residuals*, defined as the fitted values less the actual values.

Task 20 : Compute the predicted subsoil clay content for each observation, and compare it graphically with the observed subsoil clay content. •

```
> plot(fitted(lm21),Clay2,pch=20,xlab="Fitted",ylab="Observed",
+    xlim=c(5,85),ylim=c(5,85),main="Observed vs. Fitted Clay %, 0-10cm")
> abline(0,1)
> segments(fitted(lm21),Clay2,fitted(lm21),fitted(lm21))
```



Note: The `segments` method draws segments from the `(x, y)` coordinates given by the first two arguments, to the `(x, y)` coordinates given by the second pair. In this example, all four are equal-length vectors (one for each observation), so the method acts on each pair in turn. The beginning point of each segment is at (fitted, observed), while the second is at (fitted, fitted), so that the line is the vertical residual.

Q27 : *What should be the relation between the predicted and observed? Do you see any discrepancies?* *Jump to A27* •

5.7.2 Large residuals

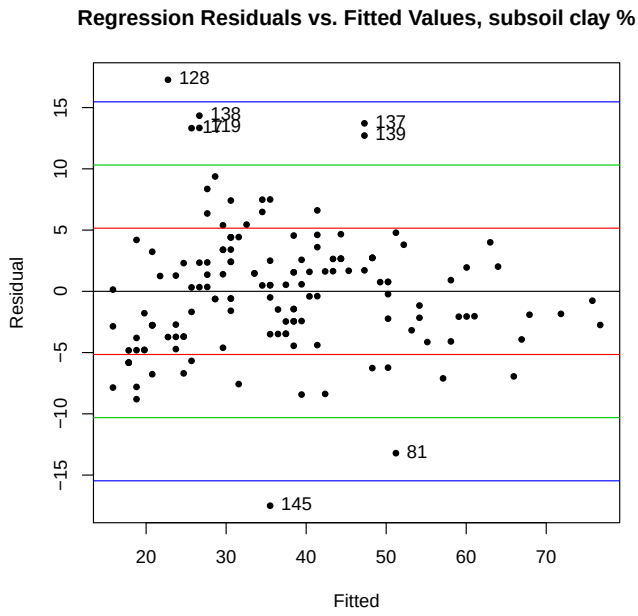
The absolute residuals, defined for observation x_i as $e_i = y_i - \hat{y}_i$ (observed less expected value) give the discrepancy of each fitted point from its observation. If any are unusually large, it may be that the observation is from a different population, or that there was some error in making or recording the observation. These residuals are interpretable directly in terms of the response variable.

Task 21 : Prepare a plot showing the residuals plotted against predicted values, along with horizontal lines showing $\pm 3, \pm 2, \pm 1$ standard deviations of the residuals. •

The plot produced by the following code also gives the observation number (index in the data frame) of each observations with unusual residuals; we find these with the `which` method. For each of these, we then display their observation number, actual topsoil and subsoil clay, fitted (predicted) subsoil clay, and the residual.

Note also the use of the `col` graphics parameter to draw the error lines in different colours depending on the number of standard deviations (`abs` method).

```
> plot(fitted(lm21), resid(lm21), pch=20, xlab="Fitted", ylab="Residual",
+    main="Regression Residuals vs. Fitted Values, subsoil clay %")
> sdres <- sd(residuals(lm21))
> for (j in -3:3) abline(h=j*sqrt(var(resid(lm21))), col=abs(j)+1)
> ix<-which(abs(resid(lm21))>2*sdres)
> text(fitted(lm21)[ix], resid(lm21)[ix], ix, pos=4)
> cbind(obs[ix,c("Clay1","Clay2")], fit=round(fitted(lm21)[ix],1),
+    resid=round(resid(lm21)[ix],1))

    Clay1 Clay2  fit resid
17     20    39 25.7  13.3
81     46    38 51.2 -13.2
119    21    40 26.7  13.3
128    17    40 22.7  17.3
137    42    61 47.3  13.7
138    21    41 26.7  14.3
139    42    60 47.3  12.7
145    30    18 35.5 -17.5

> rm(sdres, ix)
```



Q28 : *What should this relation be? Does the plot show the expected relation?* *Jump to A28* •

Q29 : *Which observations have the highest positive and negative residuals? Are these large enough to have practical significance for soil management?* *Jump to A29* •

5.7.3 Distribution of residuals

Regression residuals should be approximately normally-distributed; that is, the regression should explain the *structure* and whatever is left over (the "residue") should just be *noise*, caused by measurement errors or many small uncorrelated factors. This is precisely the theory of the normal distribution. The normality of residuals can be checked graphically and numerically.

A simple way to see the distribution of residuals is with a stem plot or histogram, using the `stem` function:

Task 22 : Make a stem plot of the residuals. •

```
> stem(residuals(lm21), scale=2)

  The decimal point is at the |

  -17 | 5
  -16 |
  -15 |
  -14 |
  -13 | 2
  -12 |
  -11 |
  -10 |
   -9 |
   -8 | 844
   -7 | 9861
   -6 | 98732
   -5 | 8887
   -4 | 8888764411
   -3 | 98777755552
   -2 | 9888877544422110
   -1 | 988765442
   -0 | 866665442
    0 | 1334555568889
    1 | 33445566666779
    2 | 033444566777777
    3 | 244468
    4 | 0244446678
    5 | 44
    6 | 456
    7 | 455
    8 | 4
    9 | 4
   10 |
   11 |
   12 | 7
   13 | 337
   14 | 3
   15 |
   16 |
   17 | 3
```

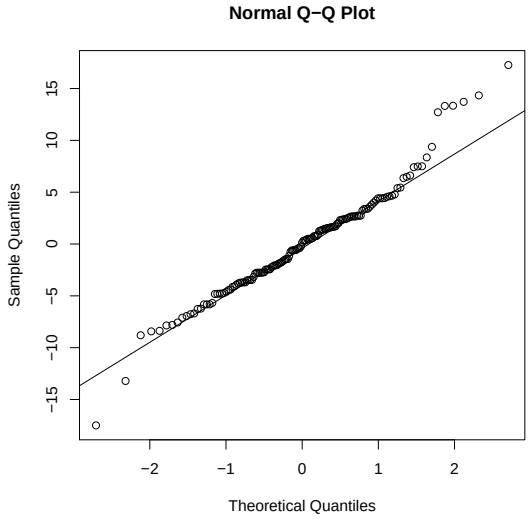
Note that without the `scale=2` optional argument to `stem`, the wide spread of residuals causes the stem plot to have one bin for each two integers, which is hard to interpret.

Q30 : *Are the residuals symmetrically-distributed? Do they appear to have a normal distribution ("bell-shaped curve")?* *Jump to A30* •

The most useful graphical tool to examine the *normality* of the residuals is the *normal quantile-quantile* ("Q-Q") plot of the regression residuals; this shows how quantiles of the residuals match to what they would be if they were taken from a normal distribution with mean 0 (by definition of "residual") and standard deviation calculated from the sample residuals.

Task 23 : Make a normal quantile-quantile ("Q-Q") plot of the regression residuals. •

```
> qqnorm(residuals(lm21))
> qqline(residuals(lm21))
```



Q31 : *Do the residuals fit the normal distribution? Where are the discrepancies, if any?* *Jump to A31* •

The hypothesis of normality can be tested with a variety of methods; these are not too useful because they often show that the null hypothesis of normality can be rejected, but the departures from normality may not be too severe.

Task 24 : Use the Shapiro-Wilk normality test of the hypothesis that the residuals are normally-distributed. •

The R function is named `shapiro.test`:

```
> shapiro.test(residuals(lm21))

        Shapiro-Wilk normality test

data:  residuals(lm21)
W = 0.969, p-value = 0.0021
```

This test computes a statistic ("W") and then compares it against a theoretical value for a normal distribution. The item in the output that can be interpreted is the p-value that rejecting the null hypothesis of normality is a Type I error.

Q32 : *With what probability can we assert that these residuals are not normally-distributed?* *Jump to A32*

•

5.7.4 Leverage *

Observations that are far from the centroid of the regression line can have a large effect on the fits. They are said to have high *leverage*, by analogy with a physical lever. These are not necessarily in error, but they should be identified and verified; in particular, it is instructive to compare the estimated regression line with and without the high-leverage observations.

The leverage is measured by the *hat value*, which measures the overall influence of a single observation on the predictions. Appendix A explains how this is derived.

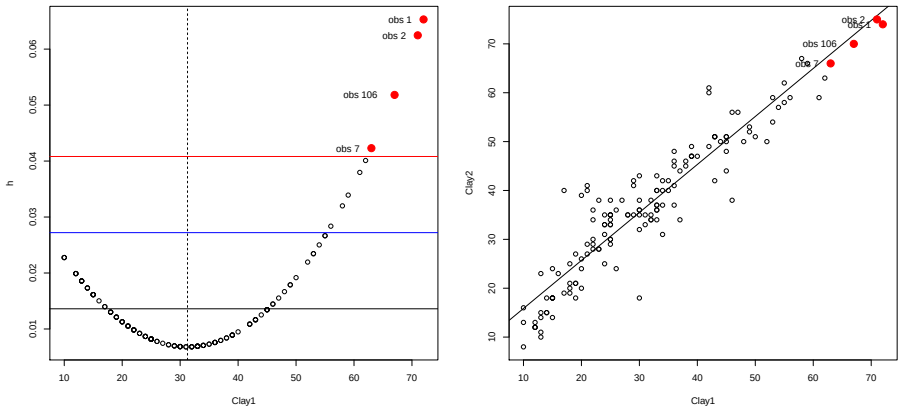
Computing the leverage with R We can find the hat values for any model with the `hatvalues` method. Values more than about three times $\bar{h}$, which is the average leverage $(k+1)/n$, where k is the number of coefficients in the model, are quite influential.

Task 25 : Find the high-leverage observations for the regression of subsoil on topsoil clay. Compare these against the highest and lowest values of the predictor. Plot the hat values against predictor value. Re-fit the model without the high-leverage observations and compare the two model coefficients. •

To compute a model with only some of the observations, use the optional `subset` argument to the `lm` method. The subset can either be inclusive (e.g. `seq[1:20]` to fit the model from the first twenty observations) or exclusive (e.g. `-c(1,5)` to use all the observations except the first and fifth).

```
> par(mfrow=c(1,2))
> h <- hatvalues(lm21)
> hi.lev <- which(h>3*mean(h)); hi.lev

  1   2   7 106
  1   2   7 106

> Clay1[hi.lev]

[1] 72 71 63 67
```

```
> (sort.list(Clay1))[(length(Clay1)-5):length(Clay1)]

[1]   3  10   7 106   2   1

> (sort.list(Clay1))[1:5]

[1]  34  39 134  82 114

> #
> plot(Clay1,h);  abline(v=mean(Clay1),lty=2)
>   abline(h=seq(1:3)*mean(h),col=c(1,4,2))
>   points(Clay1[hi.lev],hatvalues(lm21)[hi.lev],pch=20,col=2,cex=2.5)
>   text(Clay1[hi.lev],hatvalues(lm21)[hi.lev],paste("obs", hi.lev),adj=1.5)
> #
> plot(Clay1,Clay2); abline(lm21)
>   points(Clay1[hi.lev],Clay2[hi.lev],pch=20,col=2,cex=2.5)
>   text(Clay1[hi.lev],Clay2[hi.lev],paste("obs", hi.lev),adj=1.5)
> #
> lm21.2 <- lm(Clay2~Clay1,subset=-hi.lev)
> round(coefficients(lm21), 3); round(coefficients(lm21.2), 3)

(Intercept)       Clay1
      6.035       0.982

(Intercept)       Clay1
      5.718       0.994

> par(mfrow=c(1,1))
```



Q33 : *What is the relation between predictor value and its leverage?* *Jump to A33* •

Q34 : *In this particular case, do the high-leverage predictor values appear to influence the regression line?* *Jump to A34* •

Task 26 : Compare the fits of the two models, both as RMSE and R^2. The RMSE

can be computed directly from the model residuals; the R^2 as 1 − (RSS/TSS), where RSS is the *residual* sum of squares (after the model fit), and TSS is the *total* sum of squares of the predictand (here, `Clay2`), before the model fit. •

```
> sqrt(sum(residuals(lm21)^2)/length(residuals(lm21)))

[1] 5.1386

> sqrt(sum(residuals(lm21.2)^2)/length(residuals(lm21.2)))

[1] 5.1973

> 1-(sum(residuals(lm21)^2)/sum((Clay2-mean(Clay2))^2))

[1] 0.87572

> 1-(sum(residuals(lm21.2)^2)/sum((Clay2[-hi.lev]-mean(Clay2[-hi.lev]))^2))

[1] 0.85305
```

Q35 : *Does removing the high-leverage points from the dataset improve or worsen the fit in this case?* *Jump to A35* •

5.7.5 DFBETAS *

The hat values give the influence of each observation on the fitted values. Another question is the influence of each observation on the regression coefficients. This is measured by the DFBETA statistic $d[j]_i$ for the effect on coefficient β_j of omitting the ith observation, and its standardized form DFBETAS $d[j]_i{}^*$.

DFBETA is the absolute change in the coefficients with ($\hat{\beta}_j$) and without ($\hat{\beta}_{j(i)}$) the ith observation:

$$d[j]_i = \hat{\beta}_{j(i)} - \hat{\beta}_j \tag{1}$$

This can be interpreted directly: it is the absolute change in the coefficient, including the sign, if the model is re-fit without one observation. Positive values of $d[j]_i$ imply that omitting observation i would result in a larger coefficient β_j, i.e., a steeper slope.

DFBETA can be *standardized* to allow comparison among models; this is termed DFBETAS ("S"tandardized). Standardization is done by dividing the absolute difference $d[j]_i$ by the standard error $s(\hat{\beta}_{j(i)})$ of the estimate of coefficient j in the reduced regression:

$$d[j]_i{}^* = \frac{d[j]_i}{s(\hat{\beta}_{j(i)})} \tag{2}$$

A rule of thumb is that values of $d[j]_i{}^* > 2/\sqrt{n}$ are influential.

These statistics are computed with the `dfbeta` and `dfbetas` functions. The DFBETAS statistica are also included in the more general `influence.measures` function, which also reports the hat values, DFFIT, and Cook's distance.

Task 27 : Compute the DFBETA and DFBETAS statistics for the regression slope coefficient, and find the most influential observations. Show the predictor values (i.e., topsoil clay) for these. •

Note the use of the `abs` "absolute value" function, since we want to find the largest effects, both positive and negative, on the slope.

```
> # -- most influential
> dfb.lm21 <- dfbeta(lm21)[,2]
> head(ix.dfb <- order(abs(dfb.lm21), decreasing=TRUE))

[1] 128   3  81  39 127  17

> dfb.lm21[ix.dfb[1:5]]

       128          3         81         39        127
-0.0088152 -0.0075682 -0.0069635  0.0060313  0.0057800

> # -- compare with the regression coefficient and its standard error
> summary(lm21)$coefficients["Clay1",1:2]

  Estimate Std. Error
  0.982118   0.030726

> # -- standardized
> dfbs.lm21 <- dfbetas(lm21)[,2]
> # -- rule-of-thumb cutoff, depends on sample size
> (dfbetas.cutoff <- 2/sqrt(dim(obs)[1]))

[1] 0.16496

> ix.dfbs.gt <- which(abs(dfbs.lm21) > dfbetas.cutoff)
> sort(dfbs.lm21[ix.dfbs.gt])

     128        3       81       17      138       16       40
-0.29774 -0.24706 -0.23119 -0.17789 -0.17503 -0.17340  0.16751
     137      127       39
 0.17448  0.18940  0.19723

> # -- what was their predictor value?
> obs[ix.dfbs.gt,c("Clay1","Clay2")]

    Clay1 Clay2
3      61    59
16     52    50
17     20    39
39     10     8
40     13    11
81     46    38
127    13    10
```

```
128     17     40
137     42     61
138     21     41
```

Q36 : *(1) Which observation has the largest effect on the slope coefficient? (2) What is its value of the predictor? (3) How does the absolute value of the change in coefficient compare with the standard error of the slope coefficient in the full model? (4) How many observations have standardized DFBETAS larger than the rule-of-thumb cutoff for high influence?* *Jump to A36* •

5.8 Prediction

As we saw above, the best predictive equation of subsoil clay, given topsoil clay was $\text{Clay2} = 6.04 + 0.98 \cdot \text{Clay1}$, and the proportion of the variation in subsoil clay not explained by this equation was $1 - 0.8749 = 0.1251$. But what does that mean for a given prediction?

There are two sources of prediction error:

1. The uncertainty of fitting the best regression line from the available data;
2. The uncertainty in the prediction, even with a perfect regression line, because of uncertainty in the process which is revealed by the regression (i.e. the inherent noise in the process)

These correspond to the *confidence interval* and the *prediction inteveral*, respectively. Clearly, the second must be wider than the first.

The *estimation variance* depends on the variance of the regression $s^2_{Y.x}$ but also on the distance of the predictand from the centroid of the regression, $(\bar{x}, \bar{y})$. The further from the centroid, the more any error in estimating the slope of the line will affect the prediction:

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

Note that at the centroid, this reduces to $s^2_{Y.x}[(n+1)/n]$, which for large n is very close to $s^2_{Y.x}$.

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

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

Task 28 : Compute the subsoil clay predicted by the model if surface soil clay is measured as 55%, along with the confidence interval for this prediction. •

Q37 : *How much subsoil clay is expected if surface soil clay is measured as 55%?*

What is the confidence interval, based on the standard error of the regression line? Give a verbal description of the confidence interval. *Jump to A37* •

We can calculate this directly from the regression equation:

```
> round(6.0354+0.9821*55,0)

[1] 60
```

To compute the confidence interval, we could use the regression equations directly. But it is easier to use the `predict` method on the fitted model object, because this method can also compute the *standard error* of a fit, which can then be used to construct a confidence interval for that fit using the t distribution:

```
> pred <- predict(lm21,data.frame(Clay1 = 55),se.fit=T); str(pred)

List of 4
 $ fit           : Named num 60.1
  ..- attr(*, "names")= chr "1"
 $ se.fit        : num 0.845
 $ df            : int 145
 $ residual.scale: num 5.17

> round(pred$fit + qt(c(.025,.975), pred$df) * pred$se.fit, 1)

[1] 58.4 61.7
```

To predict many values (or even one), we call the `predict` method on the fitted model object with a list of values of the predictor at which to predict in a *data frame* with a predictor variable named the same as in the model.

This method also computes the *confidence interval* for the specific prediction (using the standard error of the fit and the t value computed with the model degrees of freedom), as well as the *prediction interval*, both to any confidence (default 0.95).

Task 29 : Using the `data.frame` method, make a prediction data frame from the minimum to the maximum of the data set, at 1% increments.

Using the `predict` method on the prediction data frame, compute the predicted values and the 95% *confidence interval* of the best regression, for all clay contents from the minimum to the maximum of the data set, at 1% increments. Examine the structure of the resulting object.

Using the `predict` method on the prediction data frame, compute the predicted values and the 95% *prediction interval* of the best regression, for all clay contents from the minimum to the maximum of the data set, at 1% increments. Examine the structure of the resulting object. •

```
> pframe <- data.frame(Clay1=seq(min(Clay1), max(Clay1), by=1))
> pred.c <- predict(lm21, pframe, interval="confidence", level=.95)
> str(pred.c)
```

```
 num [1:63, 1:3] 15.9 16.8 17.8 18.8 19.8 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:63] "1" "2" "3" "4" ...
  ..$ : chr [1:3] "fit" "lwr" "upr"

> pred.p <- predict(lm21, pframe, interval="prediction", level=.95)
> str(pred.p)

 num [1:63, 1:3] 15.9 16.8 17.8 18.8 19.8 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:63] "1" "2" "3" "4" ...
  ..$ : chr [1:3] "fit" "lwr" "upr"
```

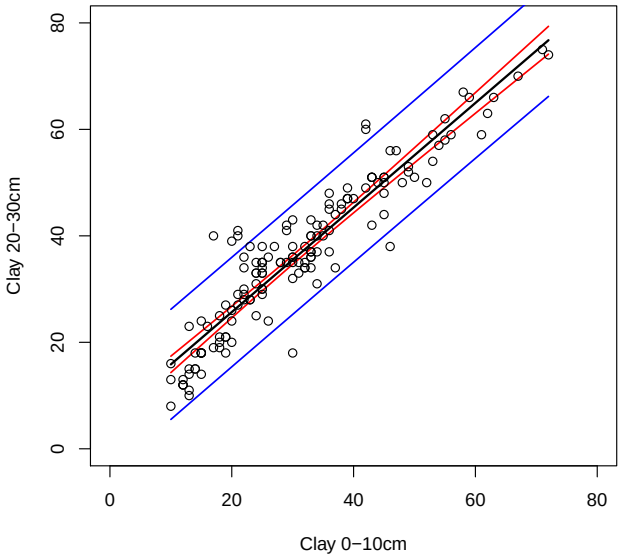
Q38 : *What are the predicted subsoil clay content, the confidence limits, and the prediction limits, for a topsoil with 55% clay? Explain in words the difference between confidence and prediction limits.* *Jump to A38* •

```
> pred.c[55+1-min(Clay1),]; pred.p[55+1-min(Clay1),]

   fit    lwr    upr
60.052 58.382 61.722

   fit    lwr    upr
60.052 49.690 70.413
```

Note: A note on this R code: the first prediction, corresponding to array position `1`, is for `min(Clay1)`, so that the prediction for 55% is at position `55+1-min(Clay1)`, i.e. `46` in the prediction array.

Task 30 : Graph the best-fit line predicted values and the 95% *confidence interval* of the best regression for the prediction frame. Also show the 95% *prediction interval*, i.e. the band in which 95% of the predicted values are expected to be. For comparison, also plot the observed values. •

```
> plot(pframe$Clay1,type="n",pred.c[,"fit"],xlab="Clay 0-10cm",
+  ylab="Clay 20-30cm",xlim=c(0,80),ylim=c(0,80))
> lines(pframe$Clay1,pred.c[,"fit"],lwd=2)
> lines(pframe$Clay1,pred.c[,"lwr"],col=2,lwd=1.5)
> lines(pframe$Clay1,pred.c[,"upr"],col=2,lwd=1.5)
> lines(pframe$Clay1,pred.p[,"lwr"],col=4,lwd=1.5)
> lines(pframe$Clay1,pred.p[,"upr"],col=4,lwd=1.5)
> points(Clay1,Clay2)
```



A note on this R code: The `predict` method returns an matrix with rows as the cases (prediction points) and columns as named dimensions, which we then access by constructions like `pred[,"upr"]`.

Q39 : *Why did we not compute the prediction for higher or lower values?* *Jump to A39* •

Q40 : *Explain the meaning of the* confidence intervals. *For what parameter of the regression are they giving the confidence? Why are these lines curved?* *Jump to A40* •

Q41 : *Explain the meaning of the* prediction intervals. *As a check, how many and what proportion of the actual data points in our sample fall outside the prediction interval? Give that there are 147 samples, how many of these would you expect to find outside the 95% prediction interval?* *Jump to A41* •

5.9 Robust regression*

Many of the problems with parametric regression can be avoided by fitting a so-called "robust" regression line. There are many variants of this, well-explained by Birkes and Dodge [2] and illustrated with S code by Venables and Ripley [35]. Here we just explore one method: `lqs` in the `MASS` package; this fits a regression to the "good" points in the dataset (as defined by some criterion), to produce a regression estimator with a high "breakdown" point. This method has several tuneable parameters; we will just use the defaults.

Task 31 : Load the `MASS` package and compute a robust regression of subsoil on surface soil clay content. Compare the fitted lines and the coefficient of determination (R^2) of this with those from the least-squares fit. •

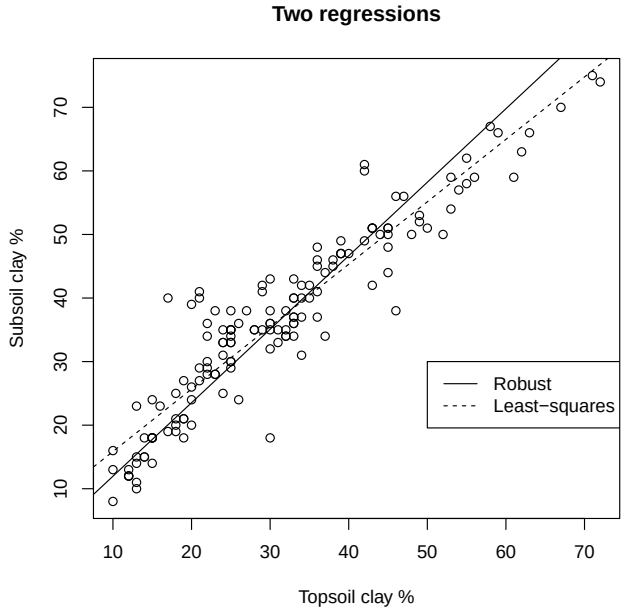
```
> require(MASS)
> lm21.r <- lqs(Clay2 ~ Clay1)
> lm21 <- lm(Clay2 ~ Clay1)
> class(lm21.r)

[1] "lqs"

> class(lm21)

[1] "lm"

> summary(lm21.r)

              Length Class      Mode
crit            1    -none-     numeric
sing            1    -none-     character
coefficients    2    -none-     numeric
bestone         2    -none-     numeric
fitted.values 147    -none-     numeric
residuals     147    -none-     numeric
scale           2    -none-     numeric
terms           3    terms      call
call            2    -none-     call
xlevels         0    -none-     list
model           2    data.frame list

> summary(lm21)

Call:
lm(formula = Clay2 ~ Clay1)

Residuals:
    Min      1Q  Median      3Q     Max
-17.499  -3.463   0.143   2.662  17.269

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.0354     1.0514    5.74  5.3e-08 ***
Clay1         0.9821     0.0307   31.96  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.17 on 145 degrees of freedom
Multiple R-squared:  0.876,        Adjusted R-squared:  0.875
F-statistic: 1.02e+03 on 1 and 145 DF,  p-value: <2e-16

> coefficients(lm21.r)

(Intercept)       Clay1
    0.40245     1.15625

> coefficients(lm21)
```

```
(Intercept)       Clay1
    6.03540     0.98212

> 1-sum(residuals(lm21.r)^2)/sum((Clay2-mean(Clay2))^2)

[1] 0.84802

> 1-sum(residuals(lm21)^2)/sum((Clay2-mean(Clay2))^2)

[1] 0.87572
```

Notice how the two `summary` methods produce very different output; this illustrates R's object-oriented methods, where objects of different classes can use the same generic methods.

Task 32 : Plot the least-squares and robust lines on the scatterplot of subsoil vs. topsoil clay. •

```
> plot(Clay2 ~ Clay1, data=obs, xlab="Topsoil clay %",
+                                                ylab="Subsoil clay %", main="Two re
> abline(lm21, lty=2)
> abline(lm21.r, lty=1)
> legend(50,30, legend=c("Robust","Least-squares"), lty=1:2)
```



Q42 : *What is the difference between the two fitted lines? Which model has the better internal fit? Why is this? What seems to be the advantage of the robust line in this case?* *Jump to A42* •

5.10 Structural Analysis*

In §5.5 we saw that the regression of two variables on each other depends on which variables is considered the predictor and which the predictand. If we are predicting, this makes sense: we get the best possible prediction. But sometimes we are interested not in prediction, but in *understanding* a relation between two variables. In the present example, we may ask what is the true relation between topsoil and subsoil clay? Here we assume that this relation has a common cause, i.e. that soil formation processes affected both the topsoil and subsoil in some systematic way, so that there is a consistent relation between the two clay contents. This so-called *structural analysis* is explained in detail by Sprent [33] and more briefly by Webster [37] and Davis ([8, pp. 214–220] and [9, pp. 218–219]).

In structural analysis we are trying to establish the best estimate for a *structural* or *law-like* relation, i.e. where we hypothesise that $y = \alpha + \beta x$, where both x and y are mathematical variables. This is appropriate when there is no need to predict, but rather to understand. This depends on the prior assumption of a true linear relation, of which we have a noisy sample.

$$\begin{aligned} X &= x + \xi \\ Y &= y + \eta \end{aligned}$$

That is, we want to observe X and Y, but instead we observe x with random error ξ and y with random error η. These errors have (unknown) variances σ^2_ξ and σ^2_η, respectively; the ratio of these is crucial to the analysis, and is symbolised as λ:

$$\lambda = \sigma^2_\eta / \sigma^2_\xi \tag{3}$$

Then the maximum-likelihood estimator of the slope, taking Y as the predictand for convention, is:

$$\hat{\beta}_{Y.X} = \frac{1}{2s_{XY}}\left\{(s^2_Y - \lambda s^2_X) + \sqrt{(s^2_Y - \lambda s^2_X)^2 + 4\lambda s^2_{XY}}\right\} \tag{4}$$

Equation 4 is only valid if we can assume that the *errors in the two variables are uncorrelated*. In the present example, it means that a large random deviation for a particular sample of the observed subsoil clay content from its "ideal" value does *not* imply anything about the random deviation of the observed topsoil clay content from *its* "ideal" value.

The problem is that we don't have any way of knowing the true error variance ratio λ, just as we have no way of knowing the true population variances, covariance, or parameters of the structural relation. We estimate the *population* variances σ^2_X, σ^2_Y and covariance σ_{XY} from the sample variances s^2_x, s^2_y and covariance s_{xy}, but there is nothing we've measured from which we can estimate the *error* variances or their ratio. However, there are several plausible methods to estimate the ratio:

- If we can assume that the two error variances are **equal**, $\lambda = 1$. This may be a reasonable assumption if the variables measure the same property (e.g. both measure clay content), use the same method for sampling and analysis, and there is an *a priori* reason to expect them to have similar variability (heterogeneity among samples).
- The two *error* variances may be estimated by the ratio of the *sample* variances: $\lambda \approx s_y^2/s_z^2$. That is, we assume that **the ratio of variability in the measured variable is also the ratio of variability in their errors**. For example, if the set of topsoil clay values in a sample is twice as variable as the set of subsoil clay values in the same sample, we would infer that the error variance is also twice as much in the subsoil, so that $\lambda = 2$. But, these are two completely different concepts! One is a sample variance and the other the variance of the error in some random process. Using this value of λ computes the *Reduced Major Axis* (RMA) [8, pp. 214-218], which is popular in biometrics.
- The variance ratio may be known from previous studies.

In the present example, we notice that $s_y^2/s_x^2 \approx 1.10$; that is, the set of subsoil samples is about 10% more variable than those from the surface soil. We could take this as the error variance ratio as well. This is not so far from 1, which is also reasonable, since both variables measure the same property.

Task 33 : Write an R function to compute $\hat{\beta}_{Y.X}$, given the structural predictand, the structural predictor, and the ratio of the error variances λ. Apply this to the structural relation between subsoil and topsoil clay, assuming equal variances, and then estimating the error variance ratio from the sample. •

```
> eqn18 <- function(y, x, lambda) {
+   a <- var(y)-lambda*var(x);
+   c <- var(x,y);
+   (a + sqrt(a^2 + 4 * lambda * c^2))/(2*c)
+ }
> eqn18(Clay2,Clay1,1)

[1] 1.053

> eqn18(Clay2,Clay1,var(Clay2)/var(Clay1))

[1] 1.0495
```

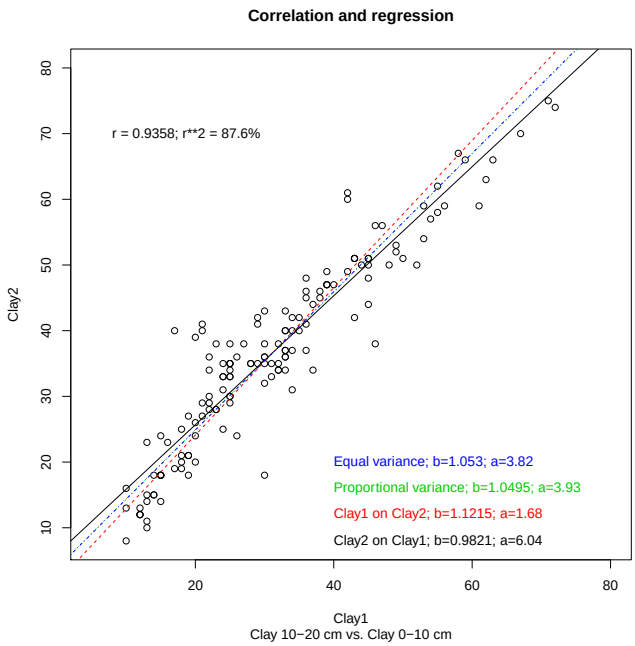
The first estimate, with $\lambda = 1$, is the *orthogonal* regression. Note that it is numerically between the slope of the regression of y on x and the inverse of the slope of the regression of x on y:

```
> lm21$coeff[2]

  Clay1
0.98212

> 1/(lm21$coeff[2])
```

```
 Clay1
1.0182

> eqn18(Clay2,Clay1,1)

[1] 1.053
```

We can plot all of these on one graph. First, we compute the coefficients directly, then we plot the graphs, and finally put some text with the computed coefficients on them.

```
> s2x <- var(Clay1); s2y <- var(Clay2); sxy <- var(Clay1,Clay2)
> byx <- sxy/s2x; ayx <- mean(Clay2)-byx*mean(Clay1)
> bxyi <- 1/(sxy/s2y); axyi <- mean(Clay2)-bxyi*mean(Clay1)
> b <- sqrt(s2y/s2x); a <- mean(Clay2)-b*mean(Clay1)
> bp <- eqn18(Clay2,Clay1,1)
> ap <- mean(Clay2)-bp*mean(Clay1)
> plot(Clay1,Clay2,xlim=c(5,80),ylim=c(8,80))
> par(adj=0.5); title("Correlation and regression","Clay 10-20 cm vs. Clay 0-10 cm")
> par(adj=0); abline(ayx,byx,col=1,lty=1)
> text(40,8,paste("Clay2 on Clay1; b=",round(byx,4),"; a=",round(ayx,2),sep=""),
+      col=1)
> abline(axyi,bxyi,col=2,lty=2)
> text(40,12,paste("Clay1 on Clay2; b=",round(bxyi,4),"; a=",round(axyi,2),sep=""),
+      col=2)
> abline(a,b,col=3,lty=3)
> text(40,16,paste("Proportional variance; b=",round(b,4),"; a=",round(a,2),sep=""),
+      col=3)
> abline(ap,bp,col=4,lty=4)
> text(40,20,paste("Equal variance; b=",round(bp,4),"; a=",round(ap,2),sep=""),
+      col=4)
> text(8,70,paste("r = ",round(cor(Clay1,Clay2),4),"; r**2 = ",
+      round((cor(Clay1,Clay2)^2)*100,1),"%",sep=""))
```



5.11 Structural Analysis by Principal Components*

This optional section presents another way to calculate structural equations in the case that error variances are equal. We use principal components analysis, which is the basic multivariate data reduction technique, discussed in more detail in 8.2. Here it is just used to compute the orthogonal axes of the two variables for which we want the structural analysis.

First, we compute the principal components for the two variables, which are put in a temporary data frame, in either order. Then we examine their *loadings* in the synthetic variables, which are the coefficients by which original observations are multiplied to produce the synthetic observations. The loadings for the first component, or *principal axis* of the new space, give the the slope of the structural regression, in this case of `Clay2` on `Clay1`; if we wanted to describe the structural relation of `Clay1` on `Clay2` we would simply invert the ratio. The intercept is computed from this slope and the centroid of the regression, as before. We also compute the proportion of the variation explained by the first axis, from the standard deviations of the two synthetic variables.

```
> pc <- princomp(cbind(Clay1,Clay2))
> pc$loadings

Loadings:
      Comp.1 Comp.2
Clay1  0.689 -0.725
Clay2  0.725  0.689

               Comp.1 Comp.2
SS loadings       1.0    1.0
Proportion Var    0.5    0.5
Cumulative Var    0.5    1.0

> b <- pc$loadings["Clay2","Comp.1"]/pc$loadings["Clay1","Comp.1"]; b

[1] 1.053

> b <- pc$loadings[2,1]/pc$loadings[1,1]; b

[1] 1.053

> a <- mean(Clay2)-b*mean(Clay1); a

[1] 3.8194

> pc$sdev

 Comp.1  Comp.2
19.8084  3.6029

> as.numeric(round(pc$sdev[1]/sum(pc$sdev)*100,1))

[1] 84.6
```

The best structural equation, $\text{Clay2} = 3.82 + 1.053 \cdot \text{Clay1}$, is the same as that computed in §5.10 for the case $\lambda = 1$, i.e. equal error variances.

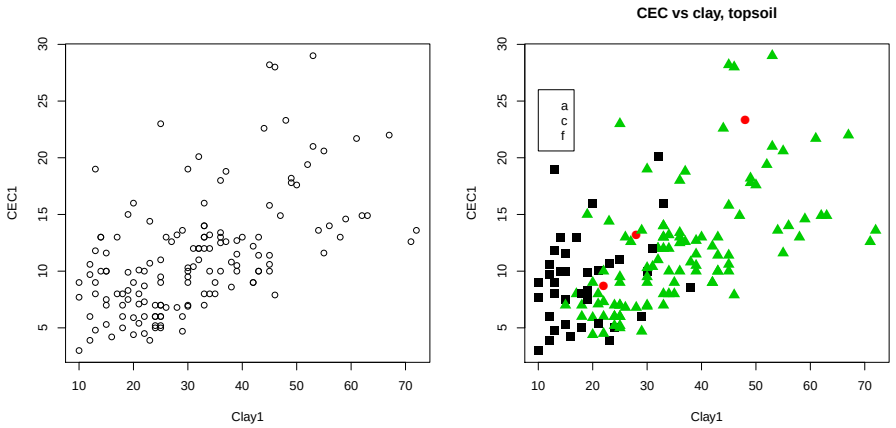
5.12 A more difficult case

The example of §5.2 – §5.11 was fairly easy to analyse. There is indeed a strong linear relation between the two variables which is only slightly affected by soil type.

In this section, by contrast, we examine a "messier" bivariate relation, to which we will return for a more satisfactory multivariate analysis in §7.2. As explained in that section, we know from theory and many detailed studies that the cation exchange capacity (CEC) of the soil depends on reactive sites where cations (such as K^+ and Ca^{++}) can be easily adsorbed and desorbed. There are such sites both on clay particles and humus; here we examine only the bivariate relation with clay.

Task 34 : Examine the relation between topsoil clay (as the predictor) and topsoil cation exchange capacity (as the predictand); first using all points and then showing the soil type. •

```
> par(mfrow=c(1,2))
> plot(Clay1,  CEC1)
> plot(Clay1,  CEC1,
+       pch=as.numeric(wrb1)+14,
+       col=as.numeric(wrb1), cex=1.5)
> title("CEC vs clay, topsoil")
> legend(10, 26, levels(wrb1),
+       pch=as.numeric(levels(wrb1))+14,
+       col=as.numeric(levels(wrb1)))
> par(mfrow=c(1,1))
```



Q43 : *Is there an apparent relation between clay and CEC? Is this consistent across clay contents? Is it linear? Do there seem to be differences between soil types?* *Jump to A43* •

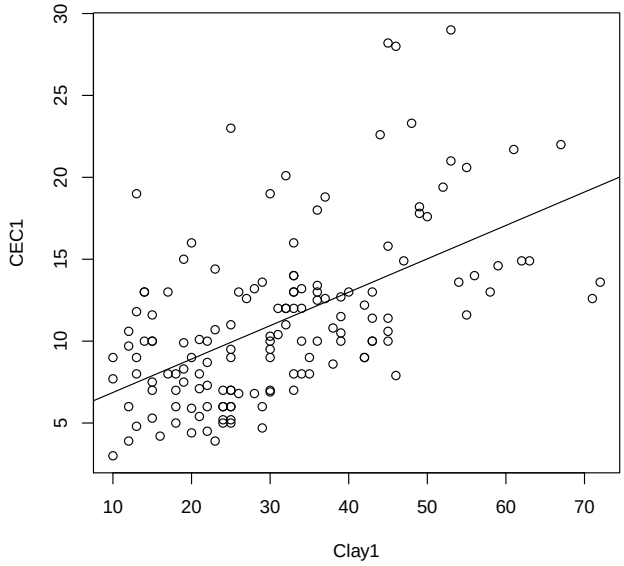
Task 35 : Compute the bivariate correlation (§5.3) of topsoil CEC and clay content. •

```
> cor.test(CEC1, Clay1)

        Pearson's product-moment correlation

data:  CEC1 and Clay1
t = 8.1, df = 145, p-value = 2.1e-13
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.43540 0.66022
sample estimates:
    cor
0.55796
```

Q44 : *How strong is the linear correlation? Compare the confidence interval with that from the correlation between topsoil and subsoil clay.* *Jump to A44* •

Task 36 : Compute and plot the bivariate regression (§5.5) of topsoil CEC on clay content. •

```
> model.cec1 <- lm(CEC1 ~ Clay1); summary(model.cec1)

Call:
lm(formula = CEC1 ~ Clay1)

Residuals:
   Min     1Q Median     3Q    Max
-6.706 -3.351 -0.645  2.201 14.196

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.8262     0.8620     5.6  1.0e-07 ***
Clay1         0.2039     0.0252     8.1  2.1e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.24 on 145 degrees of freedom
Multiple R-squared:  0.311,        Adjusted R-squared:  0.307
F-statistic: 65.5 on 1 and 145 DF,  p-value: 2.11e-13

> plot(Clay1, CEC1)
> abline(model.cec1)
```

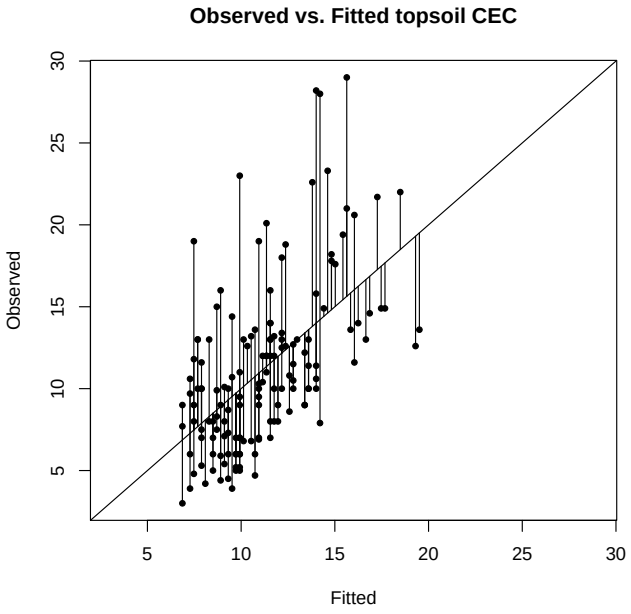


Q45 : *What is the equation of the least-squares linear regression? How much of the variability in CEC does it explain? What is the spread of the residuals (unexplained variation)?* *Jump to A45* •

Task 37 : Compute and plot the regression diagnostics (§5.7). •

```
> plot(fitted(model.cec1),CEC1,pch=20, xlab="Fitted", ylab="Observed",
+      xlim=range(CEC1),ylim=range(CEC1))
> title("Observed vs. Fitted topsoil CEC"); abline(0,1)
> segments(fitted(model.cec1),CEC1,fitted(model.cec1),fitted(model.cec1))
> #
> sdres  <-  sd(residuals(model.cec1))
> plot(fitted(model.cec1), resid(model.cec1), pch=20,
+    xlab="Fitted", ylab="Residual",
+    main="Regression Residuals vs. Fitted Values, topsoil CEC")
> #
> for (j in -3:3) abline(h=j*sqrt(var(resid(model.cec1))), col=abs(j)+1)
> ix <- which(abs(resid(model.cec1))>2*sdres)
> text(fitted(model.cec1)[ix], resid(model.cec1)[ix], ix, pos=4)
> cbind(obs[ix,c("Clay1","Clay2")], fit=round(fitted(model.cec1)[ix],1),
+    resid=round(resid(model.cec1)[ix],1))

   Clay1 Clay2  fit resid
13    48    50 14.6   8.7
23    32    35 11.4   8.7
63    13    15  7.5  11.5
73    25    35  9.9  13.1
77    44    50 13.8   8.8
78    53    54 15.6  13.4
80    45    44 14.0  14.2
81    46    38 14.2  13.8
```

```
> rm(sdres, ix)
> #
> qqnorm(residuals(model.cec1))
> qqline(residuals(model.cec1))
```



Q46 : *Are there unusual residuals? Do the residuals suggest unequal variance in the predictand throughout the range of the predictor ("heteroscedascity")?* *Jump to A46* •

We will return to this example for a more satisfactory multivariate analysis in §7.2. It turns out that much of the violation of the assumption of normal regression residuals can be accounted for by other variables.

5.13 Non-parametric correlation

Clearly, the relation between topsoil CEC and clay is not bivariate normal, so the parametric (Pearson's) correlation computed above is not a valid measure of their association. So, the Pearson's coefficient should not be reported.

The alternative is a *non-parametric* measure of bivariate association. The most common is a *rank* correlation, and the most common of these is *Spearman's* ρ, where the ranks of the two variables are correlated as with the Pearson's test.

The `rank` function returns the ranks of each observation:

```
> head(CEC1, n=10)

 [1] 13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6  7.9 14.9

> head(rank(CEC1), n=10)

 [1] 115.0  98.0 140.0  87.5 123.0 132.0 123.0 121.0  40.0 123.0
```

```
> head(Clay1, n=10)

 [1] 72 71 61 55 47 49 63 59 46 62

> head(rank(Clay1), n=10)

 [1] 147.0 146.0 142.0 137.5 128.0 130.5 144.0 141.0 126.5 143.0
```

The first paired observation is (CEC, clay) = (13.6,72); these have ranks (115,147) respectively of the 147 observations.

The Pearson's correlation of the ranks is the Spearman's ρ:

```
> cor(rank(CEC1),rank(Clay1))

[1] 0.57338

> cor(CEC1,Clay1, method="spearman")

[1] 0.57338
```

Note that in this case $\rho > r$:

```
> cor(CEC1,Clay1)

[1] 0.55796
```

5.14 Answers

A12 : *The relation is strong, linear, and positive. An increase in clay in the surface is accompanied by a proportional increase in clay in the upper subsoil.* *Return to Q12* •

A13 : *The relation looks similar; Soil type 1 (circles, black) has lower values in both layers, Soil type 3 (crosses, green) has all the high values in both layers, but it does not appear that a different line would be fitted through each class separately. There are only three samples of soil type 2 (diamonds, red).* *Return to Q13* •

A14 : *The correlation is almost certainly different from zero, since the p-value for the alternative hypothesis of no relation is almost zero. Thus there is almost certainly a relation.* *Return to Q14* •

A15 : *If this sample is representative of the population, the most likely value for the correlation of clay in the 0–10 and 10–20 cm layers, over the whole population, is estimated from this sample as 0.936. If we repeated the sampling operation (147 samples) a large number of times, in 95% of the cases we would expect to find the sample correlation coefficient between 0.912 and 0.953.*

This is a very high positive linear correlation, since the maximum is 1. *Return to Q15* •

A16 : *The smooth line has a considerably steeper slope than the "best" line at low clay*

values (till about 22%) and then a slightly shallower slope at high values. The first part of the line fits soil type 1 and the second soil type 3. *Return to Q16* •

A17 : *At low values of the parameter the line is very erratic; at high values it misses the separate "pieces" of the relation. The default value in this case gives a good visualisation.* *Return to Q17* •

A18 : `Clay2 = 6.04 + 0.98 * Clay1` *Return to Q18* •

A19 : *Subsoil clay is about 6% higher than topsoil clay on average, so that if there is no topsoil clay, we predict 6% subsoil clay. For each 1% extra topsoil clay, there is an increase in 0.98% in subsoil clay.* *Return to Q19* •

A20 : *This is given by the adjusted R-squared: 0.8749. Note that this is a bit smaller than the value computed by simply squaring the correlation coefficient:*

```
> cor(Clay2,Clay1)^2

[1] 0.87572
```

Return to Q20 •

A21 : *The total* squared *length of the lines would increase; the shortest possible squared lengths are shown here.* *Return to Q21* •

A22 : `Clay1 = -1.49 + 0.89 * Clay2` *Return to Q22* •

A23 : *Topsoil clay is about 1.5% lower than subsoil clay on average, so if there is no subsoil clay, we predict less than zero topsoil clay, which is not physically-possible. For each 1% extra subsoil clay, there is an increase in 0.89% in subsoil clay.* *Return to Q23* •

A24 : *This is given by the adjusted R-squared: 0.8749; it is exactly the same as the reverse regression.* *Return to Q24* •

A25 : *The errors are different in the different directions, leading to different east-square estimates of the best fit.* *Return to Q25* •

A26 : *The errors are different in the different directions, leading to different least-square estimates of the best fit.* *Return to Q26* •

A27 : *They should be identical, i.e. fall on the 1:1 line. Of course they are not because of error. In any case they should be symmetric about a 1:1 line (i.e. the length of the*

residual segments should be approximately equal above and below the line) throughout the range.

In this case, low predicted values are consistently too high (i.e. the observed was lower than predicted, below the line). In addition, there are several points that are quite poorly-fitted.

Return to Q27 •

A28 : *The residuals should ideally all fall on the* 0 *horizontal line; of course they do not because of error. However, in any case they should be symmetric about this line throughout the range, and have the same degree of spread.*

In this case, the low predicted values all have negative residuals (as we saw above). Also, values in the 30–40% predicted range generally have positive residuals. The spread seems about the same.

There are two residuals more than 3 standard deviations from the mean, one positive and one negative. There are five positive and one negative residual between 2 and 3 standard deviations from the mean. *Return to Q28* •

A29 : *Observation 128 has a residual of* $+17.3$*: from a topsoil content of 17% we predict 22.7% but 40% was measured; this is much higher than predicted. Observation 145 has a residual of* -17.3*: from a topsoil content of 30% we predict 35.5% but only 18% was measured; this is much lower than predicted, and indeed one of the few cases where subsoil clay is substantially lower than topsoil. Absolute residuals above about 8% clay content are indeed significant for management.* *Return to Q29* •

A30 : *The residuals are more or less symmetric about 0 but there are large "tails" in both directions with some extreme values:* $-17.5, -13.2$ *and* $+13.7, 14.3, 17.3$*. Even near 0 the distribution does not appear regular: there are too many residuals in the* -2 *and* $+2$ *bins compared to the adjacent* -1 *and* $+1$ *bins.* *Return to Q30* •

A31 : *For the most part, the residuals follow the theoretical normal distribution well: they are near the normal line and thicker near the middle. However, two low values are under-predicted (i.e. their value is below the line: these values should only be found at a lower quantile), and about five are over-predicted (i.e. their value is above the line: values this great should only be found at a higher quantile). So the tails of the residuals do not fit the normal distribution, but the bulk of the residuals do.* *Return to Q31* •

A32 : *The Shapiro-Wilk test shows that almost certainly (*$p \approx 0.002$*) we would not commit a Type I error if we reject the null hypothesis; i.e. the residuals are most likely not normally-distributed. However, this is not so serious since they are symmetrically-distributed, and the positive and negative departures from normality are somewhat balanced.* *Return to Q32*
•

A33 : *The highest leverages are the furthest from the mean of the predictor.* *Return*

to Q33 •

A34 : *No, the line that would be fitted without them seems to go right through these points.* *Return to Q34* •

A35 : *The high-leverage points are consistent with the others, so removing them produces a model with poorer fit: higher RMSE and lower* R^2.* *Return to Q35* •

A36 : *(1) Observation 128 has the largest effect on the slope coefficient: absolute change is -0.0088.*

(2) Its predictor value is 17. This low value of the predictor, if omitted, results in a lower slope coefficient. This is because the response variable here is 40, which is quite high considering the low topsoil clay concentration. So this point had quite some influence on the slope: it pulled it up at the lower end, so that omitting it would decrease the slope, hence the negative value of DFBETA.

(3) The absolute value of the change in coefficient -0.0088 is a substantial proportion of the standard error of the slope coefficient in the full model 0.0307, about -29%.

(4) A total of 10 observations have standardized DFBETAS larger than the rule-of-thumb cutoff for high influence. *Return to Q36* •

A37 : *The predicted value is*
`6.0354 + 0.9821 * 55 =  60.0509`
which rounds to `60` *(remember the precision of measurement). The confidence limits are* `round(pred$fit-qt(.975,pred$df)*pred$se.fit,1) = 58.4` *and*
`round(pred$fit+qt(.975,pred$df)*pred$se.fit,1) = 61.7`.

Return to Q37 •

A38 : *If the topsoil clay content is measured as 55%, the predicted subsoil clay content (rounded to the nearest %) is 60%; the 95% confidence limits are 58.4% and 61.7% (same as previous answer). The 95% prediction limits are 49.7% and 70.4%. The* confidence limits *are the 95% confidence of the expected value of the predictand (subsoil clay) at a predictor value of 55% topsoil clay. The* prediction limits *are the limits within which we expect 95% of all measured values of the predictand (subsoil clay) at a predictor value of 55% topsoil clay. That is, if we measured 100 locations with 55% topsoil clay, we expect that 95 of the subsoil clay contents would be between 49.7% and 70.4%.* *Return to Q38* •

A39 : *The equation is only calibrated in the range of the predictor.* *Return to Q39* •

A40 : *The confidence bands refer to the best-fit regression line.* *Return to Q40* •

A41 : *The prediction bands refer to the predicted values at each value of the predictor. In our sample, 8 of the 147 points, i.e. 5.4%, are outside these bands. This agrees very*

well with the 95% confidence interval ($0.05 \times 147 = 7.35$). *Return to Q41* •

A42 : *The robust line fits the soils with 10–50% topsoil clay better than the least-squares line. The high end of the range (about 10% of the dataset) has relatively lower subsoil clay than predicted by the robust line. The goodness-of-fit as measured by the explained sum of squares is by definition best for the least-squares fit. However, the robust fit is only a bit poorer in this sense yet fits the bulk of the data better.* *Return to Q42* •

A43 : *There is clearly a positive relation: in general, higher clay soils have higher CEC. However this is far from linear; in particular both high-clay and high-CEC soils show large discrepancies: some high-CEC soils have low clay and some high-clay soils have relatively low CEC. Soil type 1 has lower values overall but also (it appears) a steeper relation: CEC seems to increase more per unit clay increase than for soil 3.* *Return to Q43* •

A44 : *The correlation of* $r = 0.558$ *is moderate* ($R^2 = 0.311$) *but far less than for the close relation between topsoil and subsoil clay* ($r = 0.936$, $R^2 = 0.876$)*; the confidence interval is also quite wide* ($0.435 \ldots 0.660$) *compared to the two clays* ($0.912 \ldots 0.953$)*, showing the weaker relation.* *Return to Q44* •

A45 : *The least-squares line is* `CEC1 = 4.826 + 0.204 * Clay1`. *This implies that even with zero clay there would be some CEC, suggesting that there is another source of CEC than just clay. Only about 31% of the variability of CEC is explained by clay; this also suggests another source. Residuals range from* $-6.7 \ldots +14.2$ *which is a substantial spread, given that the range of CEC itself is only* $3 \ldots 29$. *Return to Q45* •

A46 : *Not only is the fit poor, but the regression residuals are not evenly-distributed. There are many more extreme positive than negative residuals, showing that high-CEC soils are poorly-modelled from clay alone; eight residuals (numbered in the residual plot) are more than two standard deviations from zero. The residuals are far from normally-distributed (see the normal Q-Q plot). However, there does not appear to be any heteroscedascity; the spread of residuals seems consistent over the range of the fitted values. Thus the linear model is not adequate and should not be used for prediction.* *Return to Q46* •

6 One-way Analysis of Variance (ANOVA)

Analysis of Variance (ANOVA) is used to determine how much of the variability in some property is explained by one or more categorical factors. As with regression, this does not necessarily imply a *causal* relation from the factors to the response. However, it does supply evidence to support a theory of causation that can be justified with a conceptual model.

The simplest ANOVA is *one-way*, where the *total* variance of the data set is compared to the *residual* variance after each observation's value is adjusted for the mean for the one factor.

In the current data set, we can ask how much the clay content varies among the four zones (variable `zone`). Clearly, the zone itself doesn't cause clay to vary, but

it is certainly reasonable that some other factor associated with the zone could be at least a partial cause. Here the zones are defined by elevation and relief, with limits corresponding to differences in parent rock and also with orographic rainfall.[4]

Because the surface soil is more subject to human disturbance and local erosion, we will model the clay in the lower subsoil (30-50 cm).

6.1 Exploratory Data Analysis

Task 38 : Visualise the clay content of the lower subsoil by zone. •

It is always a good idea to look first at the data values themselves, and then summarise or graph them. So first we sort the observations by their clay content and see which zones seem to be associated with lower or higher values. The `sort` method sorts one vector. The `order` method lists the *indices* (row or observation numbers in the data frame) that would produce this order, and these indices can be used as subscripts for another variable or the entire data frame.

```
> sort(Clay5)

  [1] 16 16 19 20 20 20 23 24 25 25 25 25 27 27 27 27 28 30 31 31 32
 [22] 32 32 32 33 33 33 33 33 33 34 34 35 35 35 36 36 37 37 37 38 38
 [43] 38 38 38 39 40 40 40 40 40 40 40 40 41 41 41 41 42 42 42 43 43
 [64] 43 43 43 43 44 44 44 44 44 44 44 45 45 45 45 45 45 45 45 45 46
 [85] 46 46 46 47 47 47 47 47 47 48 48 48 48 48 48 49 50 50 51 51 52
[106] 52 53 53 53 54 54 54 54 55 55 55 55 56 56 57 57 57 57 57 57 57
[127] 57 58 58 58 58 60 61 62 62 65 65 66 66 66 70 70 70 72 73 78 80

> order(Clay5)

  [1] 125 134  39  91 104 135 145 114  64  90 124 133  40  49  84 132
 [17] 111  63  34  82  36  46  65  66  37  41  43  50 112 127  19 126
 [33]  14  15  35  67 130  20 118 131  29  47  68 116 140 144  33  38
 [49]  51  70  74 101 123 141  21  55  56 115  57 100 117  17  26  53
 [65]  71  75 107  58  69  76  81  96 102 108  25  59  60  73  93  99
 [81] 103 110 146  30  44  92 147  88  95  98 128 138 143  23  31  32
 [97]  42  72 119  94  87 136  11  27  85  86   5  45  52  13  22  48
[113]  77  12  28  89  97  18  24   6  61  62  78  79  80 109 142  16
[129] 113 120 129  83   4  10 121  54 122   3 137 139   7   9 105   8
[145] 106   1   2

> Clay5[order(Clay5)]

  [1] 16 16 19 20 20 20 23 24 25 25 25 25 27 27 27 27 28 30 31 31 32
 [22] 32 32 32 33 33 33 33 33 33 34 34 35 35 35 36 36 37 37 37 38 38
 [43] 38 38 38 39 40 40 40 40 40 40 40 40 41 41 41 41 42 42 42 43 43
 [64] 43 43 43 43 44 44 44 44 44 44 44 45 45 45 45 45 45 45 45 45 46
 [85] 46 46 46 47 47 47 47 47 47 48 48 48 48 48 48 49 50 50 51 51 52
[106] 52 53 53 53 54 54 54 54 55 55 55 55 56 56 57 57 57 57 57 57 57
[127] 57 58 58 58 58 60 61 62 62 65 65 66 66 66 70 70 70 72 73 78 80
```

[4] caused by the mountain forcing moist warm air to rise into the cooler and less-dense atmosphere, resulting in precipitation

```
> zone[order(Clay5)]

  [1] 4 4 4 4 4 4 2 4 4 4 4 4 4 4 2 4 4 4 4 4 4 3 4 4 4 4 4 3 4 4 3 4
 [33] 1 1 4 3 3 3 3 3 3 3 3 3 2 4 4 3 3 3 3 4 3 3 3 3 3 3 3 2 3 3
 [65] 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 4 2 2
 [97] 4 2 4 3 3 2 2 3 2 3 2 2 3 1 4 2 3 2 3 2 2 2 3 1 2 2 3 3 3 2 2 2
[129] 2 2 2 2 1 1 2 2 2 1 2 2 1 2 2 2 2 2 2
Levels: 1 2 3 4
```

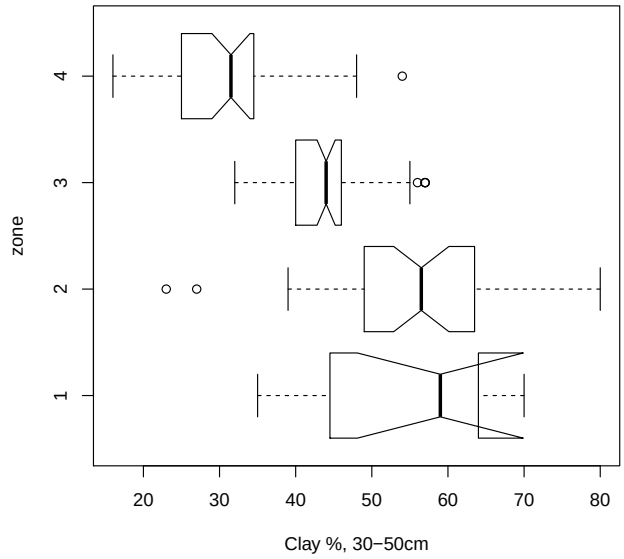
We can use the `by` method to compute any statistic for each level of a factor. Here we do it for the range. The second example illustrates the use of an anonymous function to compute the width of the range. The `function(x)` will be called with the vector of clay contents for each zone in turn, so that `max(x)` and `min(x)` will operate on this vector.

```
> by(Clay5,zone,range)

zone: 1
[1] 35 70
------------------------------------------------------
zone: 2
[1] 23 80
------------------------------------------------------
zone: 3
[1] 32 57
------------------------------------------------------
zone: 4
[1] 16 54

> by(Clay5,zone,function(x) max(x)-min(x))

zone: 1
[1] 35
------------------------------------------------------
zone: 2
[1] 57
------------------------------------------------------
zone: 3
[1] 25
------------------------------------------------------
zone: 4
[1] 38
```

Q47 : *Do there appear to be differences between zones in the* data values *and their* ranges*? If so, describe them in words.* *Jump to A47* •

Second, we can visualise this with the `boxplot` method, dividing the response variable (here, `Clay5`) by a factor (here, `zone`), using the same syntax as `lm`. In addition, if we select the `notch=T` option, this method will show whether the class *medians* are significantly different.

```
> boxplot(Clay5~zone, notch=T, horizontal=T, xlab="Clay %, 30-50cm", ylab="zone")
```



Q48 : *Do there appear to be differences between zones in the* data values *and their* ranges*? If so, describe them in words.* *Jump to A48* •

Q49 : *Are there an equal number of samples in each zone? (Hint: use the* `by` *method to divide by zones, with the* `length` *method to summarise.)* *Jump to A49* •

Q50 : *Does there appear to be a difference between zones in the* data distribution*? If so, describe it in words.* *Jump to A50* •

Task 39 : Compare the data summaries for clay contents of the lower subsoil by zone; this is the numerical confirmation of the boxplots. •

```
> by(Clay5, zone, summary)

zone: 1
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   35.0    49.2    59.0    55.0    63.0    70.0
------------------------------------------------------
zone: 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   23.0    49.5    56.5    56.0    62.8    80.0
------------------------------------------------------
zone: 3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   32.0    40.0    44.0    43.8    46.0    57.0
```

```
------------------------------------------------------
zone: 4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   16.0    25.0    31.5    31.3    34.2    54.0
```

6.2 One-way ANOVA

Based on the boxplots and descriptive statistics, it seems that the zone "explains" some of the variation in clay content over the study area. The technique for determining how much is explained by a factor is the **Analysis of Variance** (ANOVA). R's `lm` method is used for ANOVA as well as for regression; in fact it is just another form of the same linear modelling.

Task 40 : Calculate the one-way ANOVA of subsoil clay on zone, and display the ANOVA table. •

```
> lmz<-lm(Clay5~zone); summary(lmz)

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

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
```

Q51 : *How much of the total variation is explained by zones?* *Jump to A51* •

The summary for a categorical model shows the **class means**:

- The estimate on the first line, here labelled `(Intercept)` with value 55.0, is the **mean** for the first-listed class, here zone 1.
- The estimate on the second line, here labelled `zone2` with value 0.95, is the **difference** between the mean of the second-listed class; in this example the mean for zone 2 is 55.0 + 0.95 = 55.95.
- The remaining classes are computed as for the second-listed class.

We can also see this result as a classical ANOVA table, by using the `aov` method:

```
> summary(aov(lmz))

             Df Sum Sq Mean Sq F value Pr(>F)
zone          3  12390    4130    50.1 <2e-16 ***
Residuals   143  11782      82
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> coefficients(aov(lmz))

(Intercept)       zone2       zone3       zone4
     55.000       0.950     -11.159     -23.667
```

Q52 : *How likely is it that this much of the variability could have been explained by chance?* *Jump to A52* •

Q53 : *Can you determine the zone means from the ANOVA table?* *Jump to A53* •

Q54 : *From what was the reported probability value computed?* *Jump to A54* •

Q55 : *How were the variances of the model terms estimated?* *Jump to A55* •

6.3 ANOVA as a linear model*

We can see that ANOVA is just a special case of linear models by examining the *design matrices* of a regression and an ANOVA model; these are also called the *model* matrices. We'll look at observations 15 through 22, since they come from different zones.

Note for experts: the second matrix shown here is produced with the contrasts options set to `contr.treatment`; other choices of contrasts may be preferred; see the help for `?contrasts` and `?options`.

R code:

```
model.matrix(lm21)[15:22,]
model.matrix(lmz)[15:22,]
```

R console output:

```
   (Intercept) Clay1
15           1    22
16           1    52
17           1    20
18           1    33
19           1    21
20           1    22
21           1    26
22           1    38

   (Intercept) zone2 zone3 zone4
15           1     0     0     0
16           1     1     0     0
17           1     1     0     0
18           1     1     0     0
19           1     0     1     0
20           1     0     1     0
21           1     0     1     0
22           1     0     0     1
```

These are just the predictor values, for each observation, of the linear equation which `lm` is trying to fit by least-squares; each is matched with the value of the predictand for the same observation. In the first design matrix, there is an intercept and the topsoil clay content. In the second design matrix, there is an intercept (which will be fitted by the mean of the first level of the factor, i.e. zone 1; `mean(Clay5[zone ==1])`) and three *dummy variables*, representing the *deviations* of the means of the remaining three zones from the mean of zone 1. Observations from zone 1 (e.g. observation 15) have 0's for all of these; observations from other zones have a single 1, corresponding to the zone (e.g. observation 16 has a 1 for dummy variable `zone2`).

The design matrix X is then used in the unweighted least-squares estimate, given the sample values as the response vector $\mathbf{y}$:

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

which we can compute directly as:

R code:

```
X <- model.matrix(lm21)
(beta <- solve(t(X)%*%X) %*% t(X) %*% Clay5)

X <- model.matrix(lmz)
(beta <- solve(t(X)%*%X) %*% t(X) %*% Clay5)

rm(X, beta)
```

Note the use of the `%*%` operator for matrix multiplication.

R console output:

```
(Intercept) 18.7586
Clay1        0.8289

(Intercept)  55.000
zone2         0.950
zone3       -11.159
zone4       -23.667
```

These are the same coefficients as we found using `lm` for bivariate regression (§5.5) and one-way ANOVA (§6.2).

6.4 Means separation*

Now we infer that the model is highly significant, but are all the means different? For any two zones, we could use an unpaired t-test. For the entire ANOVA table, we could use Tukey's HSD if the sample was approximately balanced (which it isn't here). Another method is to do all the pairwise t-tests, but adjust for the number of comparisons; the `pairwise.t.test` method performs these tests. It may be used with different assumptions about the individual class variances (i.e. whether it is legitimate to pool them or not) and different degrees of correction.

Here are examples of the use of `pairwise.t.test`, first with no correction, then with a correction, then without the assumption of equal class variances; output has been edited for brevity:

```
> pairwise.t.test(Clay5, zone, p.adj="none")

        Pairwise comparisons using t tests with pooled SD

data:  Clay5 and zone

  1     2      3
2 0.787 -      -
3 0.001 8e-10  -
4 5e-10 <2e-16 8e-10

P value adjustment method: none

> pairwise.t.test(Clay5, zone, p.adj="holm")

        Pairwise comparisons using t tests with pooled SD

data:  Clay5 and zone

  1     2      3
2 0.787 -      -
3 0.003 3e-09  -
4 3e-09 <2e-16 3e-09

P value adjustment method: holm

> pairwise.t.test(Clay5, zone, p.adj="none",pool.sd=F)
```

```
        Pairwise comparisons using t tests with non-pooled SD

data:  Clay5 and zone

  1     2     3
2 0.855 -     -
3 0.050 2e-07 -
4 0.001 1e-15 2e-09

P value adjustment method: none

> pairwise.t.test(Clay5, zone, p.adj="holm",pool.sd=F)

        Pairwise comparisons using t tests with non-pooled SD

data:  Clay5 and zone

  1     2     3
2 0.855 -     -
3 0.099 6e-07 -
4 0.003 8e-15 1e-08

P value adjustment method: holm
```

Q56 : *What is the probability that the observed difference between zones 1 and 3 is due to chance, for all four methods?* *Jump to A56* •

Q57 : *How do the* p *values change if we adjust for multiple comparisons? How do they change if variances can not be pooled?* *Jump to A57* •

Q58 : *In the present case, is it reasonable to assume that class variances can be pooled?* *Jump to A58* •

6.5 One-way ANOVA from scratch*

It is instructive to see exactly how ANOVA works. The idea is to partition the total variance in a variable into the part attributable to some group (here the zone) and a residual.

The *unadjusted* R^2 is directly computed as:

$$R^2 = 1 - (\text{ResidualSS}/\text{TotalSS})$$

First we compute the grand mean and group means, to see that they're different. Then, we compute the total sum of squares and the residual sum of squares after subtracting the appropriate group mean. We also compute the group sum of squares, i.e. how much the groups explain, and check that this and the within-group sum of squares equals the total sum of squares. Finally, we can compute the proportion of variation explained.

```
> mean(Clay5)

[1] 44.68

> (means <- by(Clay5, zone, mean))

zone: 1
[1] 55
------------------------------------------------------
zone: 2
[1] 55.95
------------------------------------------------------
zone: 3
[1] 43.841
------------------------------------------------------
zone: 4
[1] 31.333

> (tss <- sum((Clay5 - mean(Clay5))^2))

[1] 24172

> (rss <- sum((Clay5 - means[zone])^2))

[1] 11782

> (gss <- sum(((means-mean(Clay5))^2)*by(Clay5, zone, length)))

[1] 12390

> (gss+rss - tss)

[1] 3.638e-12

> 1-(rss/tss)

[1] 0.51256
```

These computations show quite well how R operates on vectors. For example, in the computation of `gss`, the group means (`means`) is a vector of length 4; since the grand mean `mean(Clay5)` is a scalar, it is subtracted from each element of the vector, resulting in another vector of length 4; each element is squared; then since the `by` method also results in a vector of length 4 (the number of observations in each class), the multiplication is element-wise. Finally, the `sum` method sums the 4-element vector.

Note that the $R^2 = 0.513$, exactly as reported by `aov`.

6.6 Answers

A47 : *Zone 4 has most of the low clay contents (with a few in zone 2), zone 3 is medium,*

while zones 1 and 2 have the highest clay contents. Zone 2 has the widest spread. *Return to Q47* •

A48 : *There is a marked difference overall. Zones 1 and 2 are similarly high, zone 3 lower and zone 4 lowest.* *Return to Q48* •

A49 : *No, zone 1 is severely under-represented, and zone 3 has about half again as many samples as zones 2 and 4.* *Return to Q49* •

A50 : *Zone 2 has the widest range. It is mostly positively skewed, but has two boxplot outliers at the low end of the range. Zone 1 has very short boxplot tails, i.e. the box with 50% of the values covers most of the range (but this is probably an effect of the small sample size). Zone 3 has a somewhat narrower range than the others and is symmetric. Zone 4 is slightly positively skewed.* *Return to Q50* •

A51 : *About half (50.23% exactly).* *Return to Q51* •

A52 : *Less than* 2.2^{-16}, *i.e. practically* 0. *Return to Q52* •

A53 : *Yes. The intercept is the mean of the first level of the factor, here zone 1. It is 55% clay. The factor estimates for the other levels are added to this to get the corresponding zone mean. For example, zone 2 is* $55 + (-11.159) = 43.84$ *(compare to the descriptive statistics, above).* *Return to Q53* •

A54 : *From the F ratio of the variances (zone and residual) and the two degrees of freedom associated with these model terms.* *Return to Q54* •

A55 : *By the mean squared errors.* *Return to Q55* •

A56 : *For the case where variances are pooled: 0.0013 and 0.0026; that is, in all cases the two means are significantly different at p=0.01. For the case where variances are not pooled: 0.0497 and 0.0993. That is, if we don't adjust for the number of comparisons, the difference is significant at p=0.05, and the Holm correction still shows a significant difference but only at p=0.10.* *Return to Q56* •

A57 : *In both cases the* p*-values increase, i.e. it is more likely that the observed difference is just due to chance.* *Return to Q57* •

A58 : *No. From the boxplot it is clear that zone 1 is much more variable than the others, so the* `pool.sd=F` *option should be used.* *Return to Q58* •

7 Multivariate correlation and regression

In many datasets we measure several variables. We may ask, first, how are they *inter-related*? This is *multiple correlation analysis*. We may also be interested in *predicting* one variable from several others; this is *multiple regression analysis*.

7.1 Multiple Correlation Analysis

The aim here is to see how a set of variables are *inter-related*. This will be dealt with in a more sophisticated manner in *Principal Components Analysis* (§8.1) and *factor analysis* (§8.2).

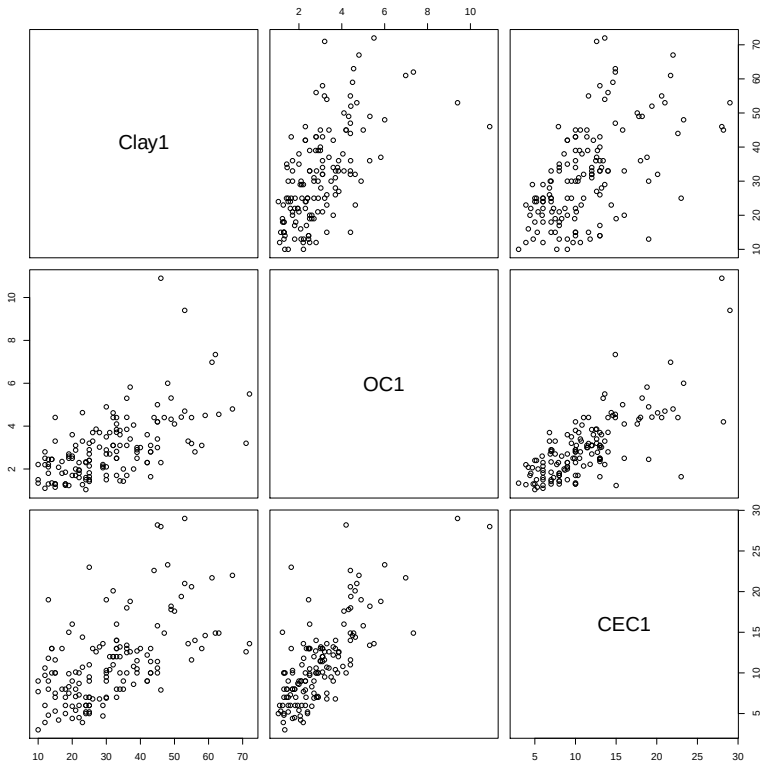
7.1.1 Pairwise simple correlations

For two variables, we used bivariate correlation analysis (§5.3). For more variables, a natural extension is to compute their *pairwise correlations* of all variables.

As explained in the next section, we expect correlations between soil cation exchange capacity (CEC), clay content, and organic carbon content.

Task 41 : Display all the bivariate relations between the three variables CEC, clay content, and organic carbon content of the 0-10cm (topsoil) layer. •

```
> pairs( ~ Clay1 + OC1 + CEC1, data=obs)
```



Q59 : *Describe the relations between the three variables.* *Jump to A59* •

The numeric strength of association is computed as for any pair of variables with a *correlation coefficient* such as Pearson's. Since these only consider two variables at a time, they are called *simple* coefficients.

Task 42 : Compute the covariances and the Pearson's correlation coefficients for all pairs of variables CEC, clay, and OC in the topsoil. •

We first must find the index number of the variables we want to plot, then we present these as a list of indices to the `cov` method:

```
> names(obs)

 [1] "e"     "n"     "elev"  "zone"  "wrb1"  "LC"    "Clay1" "Clay2"
 [9] "Clay5" "CEC1"  "CEC2"  "CEC5"  "OC1"   "OC2"   "OC5"
```

We see the target variables at positions 10, 7 and 13, so:

```
> cov(obs[c(10,7,13)])

          CEC1   Clay1     OC1
CEC1  25.9479  39.609  5.6793
Clay1 39.6092 194.213 12.5021
OC1    5.6793  12.502  2.2520

> cor(obs[c(10,7,13)])

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

Q60 : *Explain these in words.* *Jump to A60* •

7.1.2 Pairwise partial correlations

The simple correlations show how two variables are related, but this leaves open the question as to whether there are any underlying relations between the entire set. For example, could an observed strong simple correlation between variables X and Y be because both are in fact correlated to some underlying variable Z? One way to examine this is by *partial correlations*, which show the correlation between two variables after correcting for all others.

What do we mean by "correcting for the others"? This is just the correlation between the residuals of linear regressions between the two variables to be correlated and all the other variables. If the residuals left over after the regression are correlated, this can't be explained by the variables considered so far, so must be a true correlation between the two variables of interest.

For example, consider the relation between `Clay1` and `CEC1`as shown in the scatterplot and by the correlation coefficient ($r = 0.55$). These show a moderate

positive correlation. But, both of these are positively correlated to `OC1` ($r = 0.56$ and 0.74, respectively). Is some of the apparent correlation between clay and CEC actually due to the fact that soils with higher clay tend (in this sample) to have higher OC, and that this higher OC also contributes to CEC? This is answered by the partial correlation between clay and CEC, in both cases correcting for OC.

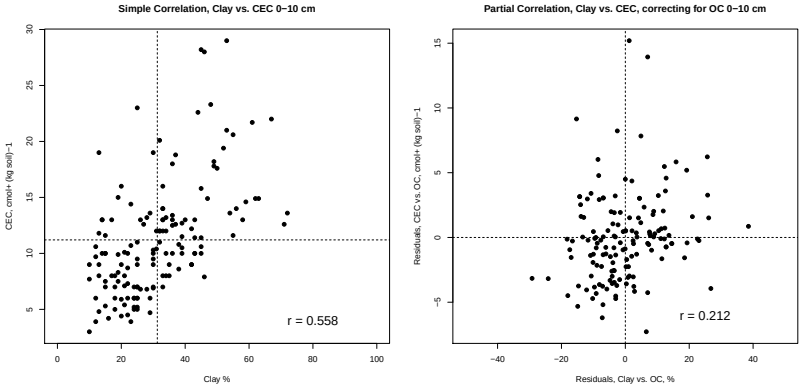
We can compute partial correlations directly from the definition, which is easy in this case with only three variables. We also recompute the simple correlations, computed above but repeated here for comparison. It's not logical (although mathematically possible) to compute the partial correlation of Clay and OC, since the "lurking" variable CEC is a result of these two, not a cause of either. So, we only consider the correlation of CEC with OC and Clay separately.

```
> cor(residuals(lm(CEC1 ~ Clay1)), residuals(lm(OC1 ~ Clay1)))

[1] 0.61538

> cor(residuals(lm(CEC1 ~ OC1)), residuals(lm(Clay1 ~ OC1)))

[1] 0.21214

> cor(CEC1, OC1)

[1] 0.74294

> cor(CEC1, Clay1)

[1] 0.55796
```

This shows that CEC is only weakly positively correlated ($r = 0.21$) to Clay after controlling for OC; compare this to the much higher simple correlation ($r = 0.56$). In other words, much of the apparent correlation between Clay and CEC can be explained by their mutual positive correlation with OC.

We can visualize the reduction in correlation by comparing the scatterplots between Clay and CEC with and without correction for OC:

```
> par(mfrow=c(1,2))
> par(adj=0.5)
> plot(CEC1 ~ Clay1, pch=20, cex=1.5, xlim=c(0,100),
+      xlab="Clay %",
+      ylab="CEC, cmol+ (kg soil)-1")
> abline(h=mean(CEC1), lty=2); abline(v=mean(Clay1), lty=2)
> title("Simple Correlation, Clay vs. CEC 0-10 cm")
> text(80, 4, cex=1.5, paste("r =",round(cor(Clay1, CEC1), 3)))
> mr.1 <- residuals(lm(CEC1 ~ OC1)); mr.2 <-residuals(lm(Clay1 ~ OC1))
> plot(mr.1 ~ mr.2, pch=20, cex=1.5, xlim=c(-50, 50),
+      xlab="Residuals, Clay vs. OC, %",
+      ylab="Residuals, CEC vs. OC, cmol+ (kg soil)-1")
> abline(h=mean(mr.1), lty=2); abline(v=mean(mr.2), lty=2)
> title("Partial Correlation, Clay vs. CEC, correcting for OC 0-10 cm")
> text(25, -6, cex=1.5, paste("r =",round(cor(mr.1, mr.2), 3)))
> par(adj=0)
> rm(mr.1, mr.2)
> par(mfrow=c(1,1))
```



The two scatterplots show that much of the apparent pattern in the simple correlation plot (left) has been removed in the partial correlation plot (right); the points form a more diffuse cloud around the centroid.

By contrast, CEC is highly positively correlated ($r = 0.62$) to OC, even after controlling for Clay (the simple correlation was a bit higher, $r = 0.74$). This suggests that OC should be the best single predictor of CEC in the topsoil; we will verify this in the next section.

The partial correlations are all smaller than the simple ones; this is because all three variables are inter-correlated. Note especially that the correlation between OC and clay remains the highest while the others are considerably diminished; this relation will be highlighted in the principal components analysis.

Simultaneous computation of partial correlations Computing partial correlations from regression residuals gets tedious for a large number of variables. Fortunately, the partial correlation can also be obtained from either the variance–covariance or simple correlation matrix of all the variables by inverting it and then standardising this inverse so that the diagonals are all 1; the off-diagonals are then the negative of the partial correlation coefficients.

Here is a small R function to do this (and give the off-diagonals the correct sign), applied to the three topsoil variables:

```
> p.cor <- function(x){
+         inv <- solve(var(x))
+         sdi <- diag(1/sqrt(diag(inv)))
+         p.cor.mat <- -(sdi %*% inv %*% sdi)
+         diag(p.cor.mat) <- 1
+         rownames(p.cor.mat) <- colnames(p.cor.mat) <- colnames(x)
+         return(p.cor.mat) }
> p.cor(obs[c(10,7,13)])

          CEC1   Clay1     OC1
CEC1  1.00000 0.21214 0.61538
Clay1 0.21214 1.00000 0.32993
OC1   0.61538 0.32993 1.00000
```

7.2 Multiple Regression Analysis

The aim here is to develop the best *predictive equation* for some predictand, given several possible predictors.

In the present example, we know that the CEC depends on reactive sites on clay colloids and humus. So it should be possible to establish a good predictive relation for CEC (the predictand) from one or both of clay and organic carbon (the predictors); we could then use this relation at sites where CEC itself has not been measured.

> Note that the type of clay mineral and, in some cases, the soil reaction are also important in modelling soil CEC; but these are similar in the sample set, so we will not consider them further.

First, we visualise the relation between these to see if the theory seems plausible in this case. This was already done in the previous section, §7.1. We saw that both predictors do indeed have some positive relation with the predictand.

To develop a predictive regression equation, we have three choices of predictors:

- Clay content
- Organic matter content
- Both Clay content and Organic matter content

The simple regressions are computed as before; the *multiple regression* with more than one predictor also uses the `lm` method, with both predictors named in the formula.

Task 43 : Compute the two simple regressions and the one multiple regression and display the summaries. Compare these with the null regression, i.e. where every value is predicted by the mean. •

```
> lmcec.null<-lm(CEC1 ~ 1); summary(lmcec.null)

Call:
lm(formula = CEC1 ~ 1)

Residuals:
   Min     1Q Median     3Q    Max
  -8.2   -3.7   -1.1    1.9   17.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    11.20       0.42    26.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.09 on 146 degrees of freedom

> lmcec.oc<-lm(CEC1 ~ OC1); summary(lmcec.oc)
```

```
Call:
lm(formula = CEC1 ~ OC1)

Residuals:
   Min     1Q Median     3Q    Max
 -7.28  -2.25  -0.21   1.58  15.19

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.671      0.630    5.82  3.6e-08 ***
OC1            2.522      0.189   13.37  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.42 on 145 degrees of freedom
Multiple R-squared:  0.552,        Adjusted R-squared:  0.549
F-statistic:  179 on 1 and 145 DF,  p-value: <2e-16

> lmcec.clay<-lm(CEC1 ~ Clay1); summary(lmcec.clay)

Call:
lm(formula = CEC1 ~ Clay1)

Residuals:
   Min     1Q Median     3Q    Max
-6.706 -3.351 -0.645  2.201 14.196

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.8262     0.8620     5.6  1.0e-07 ***
Clay1         0.2039     0.0252     8.1  2.1e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.24 on 145 degrees of freedom
Multiple R-squared:  0.311,        Adjusted R-squared:  0.307
F-statistic: 65.5 on 1 and 145 DF,  p-value: 2.11e-13

> lmcec.oc.cl<-lm(CEC1 ~ OC1 + Clay1); summary(lmcec.oc.cl)

Call:
lm(formula = CEC1 ~ OC1 + Clay1)

Residuals:
   Min     1Q Median     3Q    Max
-7.706 -2.016 -0.377  1.289 15.115

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.7196     0.7179    3.79  0.00022 ***
OC1           2.1624     0.2308    9.37  < 2e-16 ***
Clay1         0.0647     0.0249    2.60  0.01015 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.36 on 144 degrees of freedom
```

```
Multiple R-squared:  0.572,        Adjusted R-squared:  0.566
F-statistic: 96.3 on 2 and 144 DF,  p-value: <2e-16
```

Q61 : *How much of the total variability of the predictand (CEC) is explained by each of the models? Give the three predictive equations, rounded to two decimals.* *Jump to A61* •

Q62 : *How much does adding clay to the predictive equation using only organic carbon change the equation? How much more explanation is gained? Does the model summary show this as a statistically-significant increase?* *Jump to A62* •

7.3 Comparing regression models

Which of these models is "best"? The aim is to explain as much of the variation in the dataset as possible with as few predictive factors as possible, i.e. a *parsimonious* model.

7.3.1 Comparing regression models with the adjusted R^2

Compare R^2

One measure which applies to the standard linear model is the "adjusted" R^2 which decreases the apparent R^2, computed from the ANOVA table, to account for the number of predictive factors:

$$R^2_{\text{adj}} \equiv 1 - \left[\frac{(n-1)}{(n-p)} \cdot (1 - R^2) \right]$$

where n is the number of observation and p is the number of coefficients.

Q63 : *What are the adjusted R^2 in the above models? Which one is highest?* *Jump to A63* •

We can see these in the model summaries (above); they can also be extracted from the model summary:

```
> summary(lmcec.null)$adj.r.squared

[1] 0

> summary(lmcec.oc)$adj.r.squared

[1] 0.54887

> summary(lmcec.clay)$adj.r.squared

[1] 0.30657

> summary(lmcec.oc.cl)$adj.r.squared

[1] 0.56618
```

7.3.2 Comparing regression models with the AIC

Compare AIC

A more general measure, which can be applied to almost any model type, is *Akaike's Information Criterion*, abbreviated AIC. The lower value is better.

```
> AIC(lmcec.null); AIC(lmcec.oc); AIC(lmcec.clay); AIC(lmcec.oc.cl)

[1] 898.81

[1] 782.79

[1] 845.98

[1] 778.02
```

Q64 : *Which model is favoured by the AIC?* *Jump to A64* •

7.3.3 Comparing regression models with ANOVA

ANOVA, F-test

A traditional way to evaluate nested models (where one is a more complex version of the other) is to compare them in an ANOVA table, normally with the more complex model listed first. We also compute the proportional reduction in the Residual Sum of Squares (RSS):

```
> (a <- anova(lmcec.oc.cl, lmcec.clay))

Analysis of Variance Table

Model 1: CEC1 ~ OC1 + Clay1
Model 2: CEC1 ~ Clay1
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1    144 1621
2    145 2609 -1      -988 87.8 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> diff(a$RSS)/a$RSS[2]

[1] 0.3787
```

The ANOVA table shows that the second model (clay only) has one more degree of freedom (i.e. one fewer predictor), but a much higher RSS (i.e. the variability not explained by the model); the reduction is about 38% compared to the simpler model. These two estimates of residual variance can be compared with an F-test. In this case the probability that they are equal is approximately zero, so it's clear the more complex model is justified (adds information).

However, when we compare the combined model with the prediction from organic matter only, we see a different result:

```
> (a <- anova(lmcec.oc.cl, lmcec.oc))

Analysis of Variance Table
```

```
Model 1: CEC1 ~ OC1 + Clay1
Model 2: CEC1 ~ OC1
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1    144 1621
2    145 1697 -1     -76.4 6.79   0.01 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> diff(a$RSS)/a$RSS[2]

[1] 0.045004
```

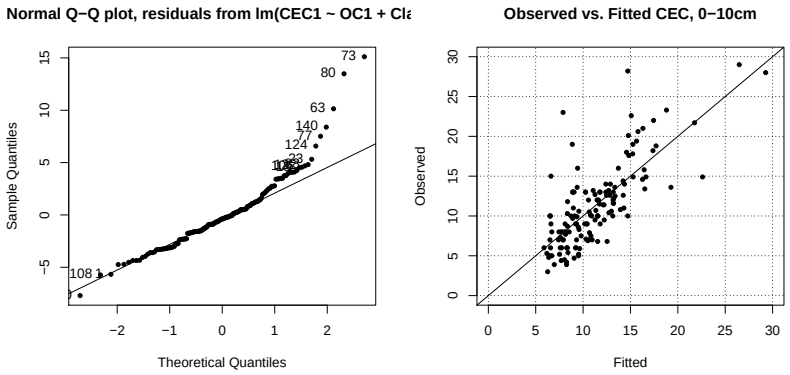
Q65 : *Which model has a lower RSS? What is the absolute and proportional difference in RSS between the combined and simple model? What is the probability that this difference is due to chance, i.e. that the extra information from the clay content does not really improve the model?* *Jump to A65* •

Regression diagnostics

Before accepting a model, we should review its diagnostics (§5.7). This provides insight into how well the model fits, and where any lack of fit comes from.

Task 44 : Display two diagnostic plots for the best model: (1) a normal quantile-quantile ("Q-Q") plot of the residuals. Identify badly-fitted observations and examine the relevant fields in the dataset, (1) predicted vs. actual topsoil CEC. •

```
> par(mfrow=c(1,2))
> tmp <- qqnorm(residuals(lmcec.oc.cl), pch=20,
+      main="Normal Q-Q plot, residuals from lm(CEC1 ~ OC1 + Clay1)")
> qqline(residuals(lmcec.oc.cl))
> diff <- (tmp$x - tmp$y)
> ### label the residuals that deviate too far from the line
> text(tmp$x, tmp$y, ifelse((abs(diff) > 3), names(diff), ""), pos=2)
> rm(tmp,diff)
> ### observed vs. fitted
> #
> plot(CEC1 ~ fitted(lmcec.oc.cl), pch=20,
+    xlim=c(0,30), ylim=c(0,30),
+    xlab="Fitted",ylab="Observed",
+    main="Observed vs. Fitted CEC, 0-10cm")
> abline(0,1); grid(col="black")
> par(mfrow=c(1,1))
```



Q66 : *Are the residuals normally distributed? Is there any apparent explanation for these poorly-modelled observations?* *Jump to A66* •

7.4 Stepwise multiple regression*

In the previous section, we examined several models individually, using our expert judgement to decide which predictors to use, and in which order. Another approach is to let R try out a large number of possible equations and select the "best" according to some criterion. One method for this is *stepwise* regression, using the `step` method.

The basic idea of `step` is to specify an initial model object, as with `lm`, and then a *scope* which specifies how variables in the full model should be added or subtracted; in the simplest case we do not specify a scope and `step` tries to eliminate all variables, one at a time, until no more can be eliminated without increasing the AIC, explained above.

We will illustrate this with the problem of predicting subsoil clay (difficult to sample) from the three topsoil parameters.

Task 45 : Set up a model to predict subsoil clay from all three topsoil variables (clay, OM, and CEC) and use `step` to see if all three are needed. •

```
> # let stepwise pick the best from a full model
> lms <- step(lm(Clay2 ~ Clay1 + CEC1 + OC1))

Start:  AIC=461.91
Clay2 ~ Clay1 + CEC1 + OC1

        Df Sum of Sq   RSS AIC
<none>                3224 462
- OC1    1        81  3305 464
- CEC1   1       179  3403 468
- Clay1  1     21078 24301 757
```

In this case we see that the full model has the best AIC (461.91) and removing any of the factors increases the AIC, i.e. the model is not as good. However,

removing either `OC1` or `CEC1` doesn't increase the AIC very much (only to 468), so although statistically valid they are not so useful.

An example with more predictors shows how variables are eliminated.

Task 46 : Set up a model to predict CEC in the 30-50 cm layer from all three variables (clay, OM, and CEC) for the two shallower layers, and use `step` to see if all six are needed. Note: this model could be applied if only the first two soil layers were sampled, and we wanted to predict the CEC value of the third layer. •

```
> lms <- step(lm(Clay5 ~ Clay1 + CEC1 + OC1 + Clay2 + CEC2 + OC2, data=obs))

Start:  AIC=420.7
Clay5 ~ Clay1 + CEC1 + OC1 + Clay2 + CEC2 + OC2

        Df Sum of Sq  RSS AIC
- CEC1   1         1 2339 419
- OC1    1         9 2347 419
- OC2    1        12 2350 419
- Clay1  1        27 2365 420
<none>               2338 421
- CEC2   1        48 2387 422
- Clay2  1      1764 4102 501

Step:  AIC=418.75
Clay5 ~ Clay1 + OC1 + Clay2 + CEC2 + OC2

        Df Sum of Sq  RSS AIC
- OC1    1        11 2350 417
- OC2    1        12 2350 417
- Clay1  1        31 2370 419
<none>               2339 419
- CEC2   1        76 2415 421
- Clay2  1      1966 4305 506

Step:  AIC=417.43
Clay5 ~ Clay1 + Clay2 + CEC2 + OC2

        Df Sum of Sq  RSS AIC
- OC2    1         5 2355 416
- Clay1  1        21 2371 417
<none>               2350 417
- CEC2   1        67 2417 420
- Clay2  1      2294 4644 516

Step:  AIC=415.77
Clay5 ~ Clay1 + Clay2 + CEC2

        Df Sum of Sq  RSS AIC
<none>               2355 416
- Clay1  1        36 2392 416
- CEC2   1        62 2417 418
- Clay2  1      2311 4666 514
```

The original AIC (with all six predictors) is 420.7; `step` examines all the variables and decides that by eliminating `CEC1` (a topsoil property) the AIC is most improved.

The AIC is now 418.75; `step` examines all the remaining variables and decides that by eliminating `OC1` the AIC is most improved; again a topsoil property is considered unimportant.

The AIC is now 417.43; `step` examines all the remaining variables and decides that by eliminating `OC2` the AIC is most improved.

The AIC is now 415.77 and all three remaining variables must be retained, otherwise the AIC increases. The final selection includes both clay measurements (0-10 and 10-20 cm) and the CEC of the second layer.

Notice from the final output that `Clay1` could still be eliminated with very little loss of information, which would leave a model with two properties from the second layer to predict the clay in the subsoil; or `CEC2` could be eliminated with a little more loss of information; this would leave the two overlying clay contents to predict subsoil clay. Either of these alternatives would be more parsimonious in terms of interpretation, although statistically just a bit weaker than the final model discovered by `step`.

7.5 Combining discrete and continuous predictors

In many datasets, including this one, we have both *discrete factors* (e.g. soil type, agro-ecological zone) and *continuous variables* (e.g. topsoil clay) which we show in one-way ANOVA and univariate regression, respectively, to be useful predictors of some continuous variable (e.g. subsoil clay). The discussion of the design matrix and linear models (§6.3) showed that both one-way ANOVA on a factor and univariate regression on a continuous predictor are just a cases of linear modelling. Thus, they can be combined in a multiple regression.

Task 47 : Model the clay content of the 20-50 cm layer from the agro-ecological zone and measured clay in the topsoil (0-10 cm layer), first separately and then as an additive model. •

```
> lm5z <- lm(Clay5 ~ zone);  summary(lm5z)

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 ***
```

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

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

> lm51 <- lm(Clay5 ~ Clay1); summary(lm51)

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

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

> lm5z1 <- lm(Clay5 ~ zone + Clay1);  summary(lm5z1)

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

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
```

Note the use of the + in the model specification. This specifies an *additive* model, where there is one regression line (for the *continuous* predictor) which is displaced vertically according to the mean value of the *discrete* predictor. This is sometimes called *parallel regression*. It hypothesizes that the only effect of the discrete predictor is to adjust the mean, but that the relation between the contin-

uous predictor and the predictand is then the same for all classes of the discrete predictor. Below (§7.8) we will investigate the case where we can not assume parallel slopes.

Q67 : *How much of the variation in subsoil clay is explained by the zone? by the topsoil clay? by both together? Is the combined model better than individual models? How much so?* *Jump to A67* •

Q68 : *In the parallel regression model (topsoil clay and zone as predictors), what are the differences in the means between zones? What is the slope of the linear regression, after accounting for the zones? How does this compare with the slope of the linear regression not considering zones?* *Jump to A68* •

Q69 : *Are all predictors in the combined model (topsoil clay and zone as predictors) asignificant? (Hint: look at the probability of the* t*-tests.)* *Jump to A69* •

Diagnostics We examine the residuals to see if any points were especially badly-predicted and if the residuals fit the hypothesis of normality.

Task 48 : Make a stem plot of the residuals. •

```
> stem(residuals(lm5z1))

  The decimal point is at the |

  -24 | 1
  -22 |
  -20 |
  -18 |
  -16 |
  -14 |
  -12 |
  -10 | 540
   -8 | 77104
   -6 | 10099662
   -4 | 888539854322
   -2 | 8655321009876110
   -0 | 986665432211098766655532 1
    0 | 00122334445679023444466688889
    2 | 0334488900122333345568
    4 | 0336800058
    6 | 35792244
    8 | 5
   10 | 11188
   12 | 49
```

Q70 : *Are the residuals normally-distributed? Are there any particularly bad values?* *Jump to A70* •

Clearly there are some points that are less well-modelled.

Task 49 : Display the records for these poorly-modelled points and compare their subsoil clay to the prediction. •

```
> res.lo <- which(residuals(lm5z1) < -12)
> res.hi <- which(residuals(lm5z1) > 9)
> obs[res.lo, ]

         e      n elev zone wrb1  LC Clay1 Clay2 Clay5 CEC1 CEC2 CEC5
145 695098 328237  547    2    f OCA    30    18    23    7    6    7
    OC1 OC2 OC5
145 1.5 0.8 0.8

> predict(lm5z1)[res.lo]

   145
47.086

> obs[res.hi, ]

         e      n elev zone wrb1  LC Clay1 Clay2 Clay5 CEC1 CEC2 CEC5
9   681230 311053  600    2    f  FV    46    56    70  7.9  5.7  4.5
27  679242 338073  360    3    a  FV    24    35    51  5.0  5.4 13.1
38  671039 336819  130    4    a OCA    13    23    40  4.8  3.4  3.2
42  667325 334883  243    4    a  FV    23    38    48  3.9  4.2  4.9
119 666452 337405  134    4    a  BF    21    40    48  5.4  2.6  7.5
128 699567 328185  630    2    f MCA    17    40    47  8.0  8.0  8.0
137 698928 328368  640    2    f  FV    42    61    66  9.0  9.0  8.0
139 695014 328757  560    2    f  FV    42    60    66  9.0  8.0  8.0
     OC1  OC2 OC5
9   2.30 1.36 0.9
27  1.04 0.52 0.5
38  1.30 0.34 0.2
42  1.27 0.58 0.5
119 2.00 0.60 0.4
128 1.80 0.90 0.8
137 2.30 1.30 1.0
139 2.30 1.20 1.0

> predict(lm5z1)[res.hi]

     9     27     38     42    119    128    137    139
58.856 39.229 28.228 35.583 34.112 37.524 55.913 55.913
```

Q71 : *What are the predicted and actual subsoil clay contents for the highest and lowest residuals? What is unusual about these observations?* *Jump to A71* •

7.6 Diagnosing multi-colinearity

Another approach to reducing a regression equation to its most parsiminious form is to examine the relation between the predictor variables and the predictand for *multi-collinearity*, that is, the degree to which they are themselves linearly related in the multiple regression. In the extreme, clearly if two variables are perfectly related, one can be eliminated, as it can not add information as a predictor.

This was discussed to some extent in §7.1 "Multiple correlation", but it was not clear which of the correlated variables to discard, because the predictand was not included in the analysis. For this we use the **Variance Inflation Factor** (VIF), which measures the effect of a set of explanatory variables (predictors) on the *variance* of the coefficient of another predictor, in the multiple regression equation including all predictors, i.e. how much the variance of an estimated regression coefficient is increased because of collinearity. The square root of the VIF gives the increase in the standard error of the coefficient in the full model, compared with what it would be if the target predictor were uncorrelated with the other predictors. Fox [12] has a good discussion, including a visualization.

In the standard multivariate regression:

$$Y = \sum_{0}^{k} \beta_k X_k + \varepsilon,\ X_0 = 1 \tag{5}$$

solved by ordinary least-squares, the sampling variance of an estimated regression coefficient $\hat{\beta}_j$ can be expressed as:

$$\mathrm{var}(\hat{\beta}_j) = \frac{s^2}{(n-1)s_j^2} \cdot \frac{1}{1-R_j^2} \tag{6}$$

where:

s^2 : is the estimated error variance of the residuals of the multiple regression;

s_j^2 : is the sample variance of the target variable;

R_j^2 : is the multiple coefficient of determination for the regression of the target variable X_j on the other predictors.

The **left-hand multiplicand** applies also in a single-predictor regression: it measures the imprecision of the fit compared to that of the predictor. A larger overall error variance of the regression, s^2, will, of course, always lead to a higher variance in the regression coefficient, while a larger number of observations n and a larger variance s_j^2 of the target variable will both lower the variance in the regression coefficient.

The **right-hand multiplicand**, $1/(1-R_j^2)$ applies only in multiple regression. This is the VIF: it multiplies the variance of the regression coefficient by a factor that will be larger as the multiple correlation of a target predictor with the other predictors increases. Thus the VIF increases as the target predictor does not add much information to the regression.

The VIF is computed with the `vif` function of John Fox's `car` package [13].

Task 50 : Load the `car` package and compute the VIF of the six predictors. •

```
> require(car)
> vif(lm(Clay5 ~ Clay1 + CEC1 + OC1 + Clay2 + CEC2 + OC2, data=obs))

  Clay1    CEC1     OC1   Clay2    CEC2     OC2
12.8391  4.7712  4.0944 10.3882  3.5531  3.0349
```

There is no test of significance or hard-and-fast rule for the VIF: however many authors consider $VIF \geq 5$ as a caution and $VIF \geq 10$ as a definite indication of multicolinearity. Note that this test does not tell *which* variables, of the set, each variable with a high VIF is correlated with. It could be with just one or with several taken together.

Q72 : *According to the* VIF ≥ 10 *criterion, which variables are highly correlated with the others?* *Jump to A72* •

Task 51 : Re-compute the VIF for the multiple regression without these variables, each taken out separately. •

```
> vif(lm(Clay5 ~ Clay1 + CEC1 + OC1 + CEC2 + OC2, data=obs))

 Clay1   CEC1    OC1   CEC2    OC2
2.5927 4.2208 4.0916 3.3218 3.0214

> vif(lm(Clay5 ~ Clay2 + CEC1 + OC1 + CEC2 + OC2, data=obs))

 Clay2   CEC1    OC1   CEC2    OC2
2.0978 4.5034 4.0277 3.5256 2.9037
```

Q73 : *According to the* VIF >= 10 *criterion, which variables in these reduced equations are highly correlated with the others? What do you conclude about the set of variables?* *Jump to A73* •

Since either `Clay1` or `Clay2` can be taken out of the equation, we compare the models, starting from a reduced model with each one taken out, both as full models and models reduced by backwards stepwise elimination:

First, eliminating `Clay2`:

```
> AIC(lm(Clay5 ~ Clay1 + CEC1 + OC1 + CEC2 + OC2, data=obs))

[1] 920.5

> AIC(step(lm(Clay5 ~ Clay1 + CEC1 + OC1 + CEC2 + OC2, data=obs), trace=0))

[1] 916.16
```

Second, eliminating `Clay1`:

```
> AIC(lm(Clay5 ~ Clay2 + CEC1 + OC1 + CEC2 + OC2, data=obs))
```

```
[1] 839.57

> AIC(step(lm(Clay5 ~ Clay2 + CEC1 + OC1 + CEC2 + OC2, data=obs) , trace=0))

[1] 835.2
```

Q74 : *Which of the two variables with high VIF in the full model should be eliminated?* *Jump to A74* •

Task 52 : Compute a reduced model by backwards stepwise elimination, starting from the full model with this variable eliminated. •

```
> (lms.2 <- step(lm(Clay5 ~ Clay2 + CEC1 + OC1 + CEC2 + OC2, data=obs)))

Start:  AIC=420.4
Clay5 ~ Clay2 + CEC1 + OC1 + CEC2 + OC2

        Df Sum of Sq   RSS AIC
- CEC1   1         5  2370 419
- OC1    1         5  2371 419
- OC2    1        22  2387 420
<none>                2365 420
- CEC2   1        56  2421 422
- Clay2  1     10782 13148 671

Step:  AIC=418.69
Clay5 ~ Clay2 + OC1 + CEC2 + OC2

        Df Sum of Sq   RSS AIC
- OC1    1         1  2371 417
- OC2    1        20  2390 418
<none>                2370 419
- CEC2   1        67  2437 421
- Clay2  1     11653 14023 678

Step:  AIC=416.75
Clay5 ~ Clay2 + CEC2 + OC2

        Df Sum of Sq   RSS AIC
- OC2    1        21  2392 416
<none>                2371 417
- CEC2   1        66  2437 419
- Clay2  1     11876 14247 678

Step:  AIC=416.03
Clay5 ~ Clay2 + CEC2

        Df Sum of Sq   RSS AIC
<none>                2392 416
- CEC2   1        47  2439 417
- Clay2  1     15687 18078 711

Call:
```

```
lm(formula = Clay5 ~ Clay2 + CEC2, data = obs)

Coefficients:
(Intercept)        Clay2         CEC2
     14.519        0.861       -0.199
```

Q75 : *What is the final model? What is its AIC? How do these compare with the model found by stepwise regression, not considering the VIF criterion?* *Jump to A75* •

Another approach is to compute the stepwise model starting from a full model, and then see the VIF of the variables retained in that model.

Task 53 : Compute the VIF for the full stepwise model. •

The `vif` function can be applied to a model object; in this case `lms`, computed above:

```
> vif(lms)

 Clay1  Clay2   CEC2
8.3567 8.0790 1.5327
```

Q76 : *What is the multi-colinearity in this model?* *Jump to A76* •

This again indicates that the two "clay" variables are highly redundant, and that eliminating one of them results in a more parsimonious model. Which to eliminate is evaluated by computing both reduced models and comparing their AIC.

Task 54 : Compute the AIC of this model, with each of the highly-correlated variables removed. •

We specify the new model with the very useful `update` function. This takes a model object and adjusts it according to a new formula, where existing terms are indicated by a period ('.').

```
> AIC(lms)

[1] 834.94

> AIC(update(lms, . ~ . - Clay1))

[1] 835.2

> AIC(update(lms, . ~ . - Clay2))

[1] 933.44
```

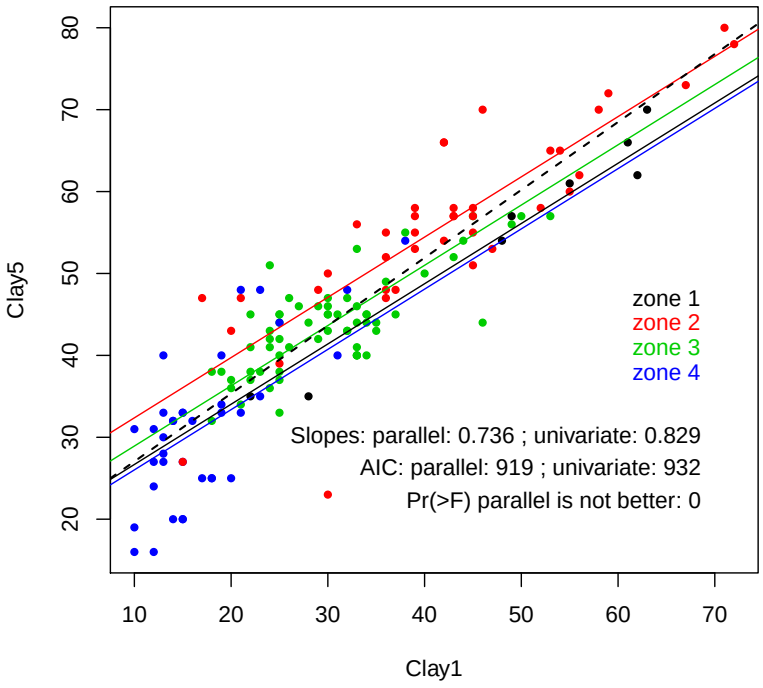
Q77 : *Which of the two "clay" variables should be eliminated? How much does this change the AIC?* *Jump to A77* •

7.7 Visualising parallel regression*

In parallel regression (additive effects of a continuous and discrete predictor) there is only one regression line, which is displaced up or down for each class of the discrete predictor. Even though there are two predictors, we can visualize this in a 2D plot by showing the displaced lines.

Task 55 : Plot subsoil vs. topsoil clay, with the observations coloured by zone. Add the parallel regression lines from the combined model, in the appropriate colours, and the univariate regression line. •

```
> # scatterplot, coloured by zone
> plot(Clay5 ~ Clay1, col=as.numeric(zone), pch=20)
> # zone 1
> abline(coefficients(lm5z1)["(Intercept)"] , coefficients(lm5z1)["Clay1"])
> # zone 2
> for (iz in 2:4) {
+    abline(coefficients(lm5z1)["(Intercept)"]
+     +  coefficients(lm5z1)[iz]
+    , coefficients(lm5z1)["Clay1"], col=iz) }
> # univariate line
> abline(lm51, lty=2, lwd=1.5)
> # legend
> text(70, 30, pos=2,
+   paste("Slopes: parallel:",
+   round(coefficients(lm5z1)["Clay1"],3),
+   "; univariate:",
+   round(coefficients(lm51)["Clay1"],3)));
> text(70, 26, pos=2,
+   paste("   AIC: parallel:", floor(AIC(lm5z1)),
+   "; univariate:", floor(AIC(lm51))));
> text(70, 22, pos=2,
+   paste("Pr(>F) parallel is not better:",
+   round(anova(lm5z1,lm51)$"Pr(>F)"[2],)))
> for (iz in 1:4) { text(65, 50-(3*iz), paste("zone",iz), col=iz) }
```



Note the use of the `coefficients` method to extract the vector of fitted coefficients, which can be accessed by name or position.

Q78 : *How well do the four parallel lines appear to fit the corresponding points, i.e. the points from the corresponding zone?* *Jump to A78* •

7.8 Interactions*

Both topsoil clay and agro-ecological zone can predict subsoil clay to some extent. Combined as an *additive* model, they do better than each separately. But this leaves the question of whether they are completely independent. In this case, we may ask if the *slope* of the regression of subsoil on topsoil clay is different.

Task 56 : Model the clay content of the 20-50 cm layer from the agro-ecological zone and measured clay in the topsoil (0-10 cm layer) as a additive model *with interactions*. •

To express an interaction between model terms, we use `*` instead of `+` in the model formula:

```
> lm51.z <- lm(Clay5 ~ Clay1 * zone)
> summary(lm51.z)

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

```
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 *
Clay1         0.8343     0.1265    6.59  8.2e-10 ***
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:zone2  -0.0955     0.1411   -0.68    0.500
Clay1:zone3  -0.2703     0.1513   -1.79    0.076 .
Clay1:zone4   0.2471     0.1877    1.32    0.190
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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
```

Q79 : *How much of the variation in subsoil clay is explained by this model? Is it better than the additive model?* *Jump to A79* •

Q80 : *Are all predictors in the combined model (topsoil clay and zone as predictors) significant? For the predictors also present in the additive model (i.e. zone and clay separately, not their interaction) are the same ones significant, and to the same degree?* *Jump to A80* •

Of most interest are the *interaction terms* in the model summary. In this model, these tell us if the relation between topsoil and subsoil clay is the same in all zones. This was the assumption of parallel (additive) regression (§7.5); but if there are interactions, there is not only one slope, but different slopes for each level of the classified predictor (here, zone).

Q81 : *Is there evidence that the relation between topsoil and subsoil clay is different in some of the zones? If so, at what significance level (i.e. probability of Type I error)?* *Jump to A81* •

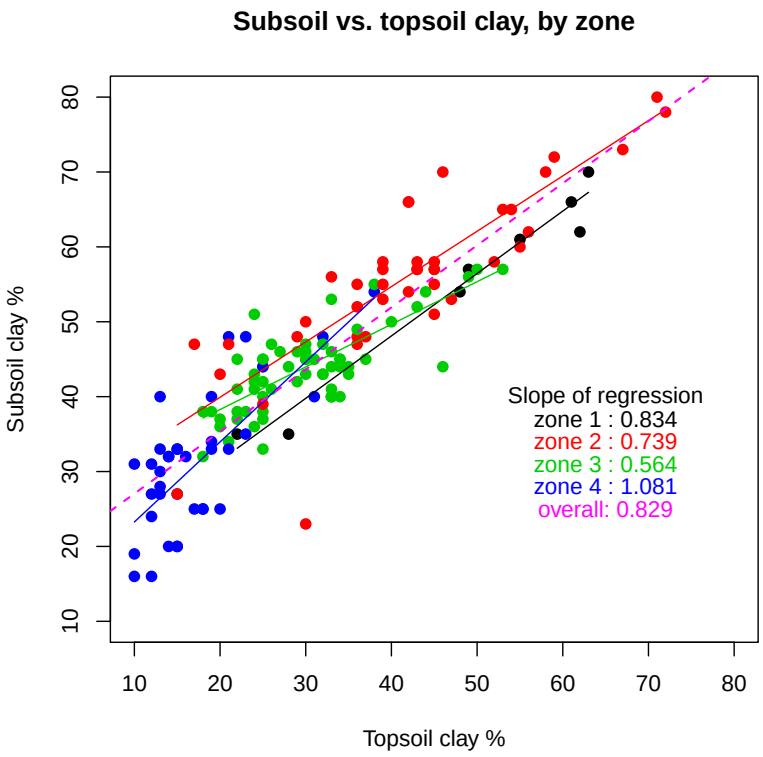
Task 57 : Visualise this by plotting the different regressions of subsoil on topsoil clay, by zone. •

To do this, we use the `subset` optional argument to the `lm` method to select just some observations, in this case, those in a zone. We plot each regression and its associated points in different colours.

Note: This code uses a "trick" to plot each regression only in the range of its subset

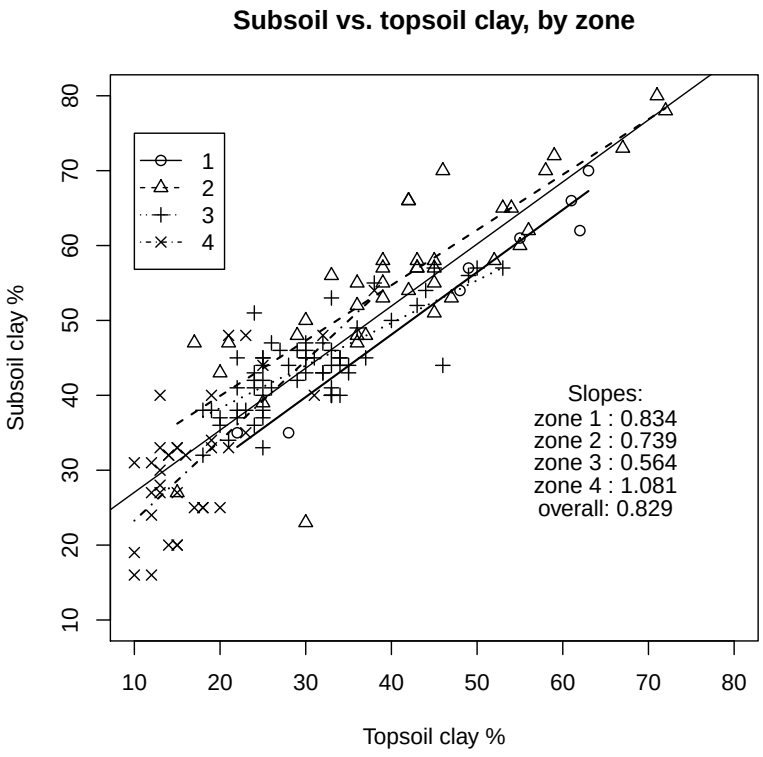
(zone). The `abline` method draws a line for the whole range of the plot, and can't be limited to a range. We use the `loess` "local polynomial regression fitting" function on the subset, selected with the `subset` argument, with a very large `span` argument to force a straight line (rather than a locally-adjusted polynomial). This returns a set of fitted values which only cover the span of the data. We plot this with the usual `lines` function, but only use the minimum and maximum fitted points (i.e. the end points of the fitted line); otherwise the line becomes too thick.

```
> plot(Clay1, Clay5, xlim=c(10,80), ylim=c(10,80), pch=20,
+     cex=1.5, col=as.numeric(zone),xlab="Topsoil clay %", ylab="Subsoil clay %");
> title("Subsoil vs. topsoil clay, by zone");
> text(65, 40, "Slope of regression");
> for (z in 1:4) {
+   m <- lm(Clay5 ~ Clay1, subset=(zone==z));
+   text(65, 40-(3*z), paste("zone",z,":",round(coefficients(m)[2], 3)),
+           col=z);
+   m.l <- loess(Clay5 ~ Clay1, subset=(zone==z), span=100);
+   lines(y=c(min(m.l$fitted), max(m.l$fitted)),
+           x=c(min(m.l$x), max(m.l$x)), col=z);
+ };
> m <- lm(Clay5 ~ Clay1);
> abline(m, col=6, lwd=1.5, lty=2);
> text(65, 25, paste("overall:", round(coefficients(m)[2], 3)), col=6);
> rm(m, m.l, z)
```



With the lines covering only part of the data, and the obviously different slopes, a black-and-white graph with different point symbols may be a cleaner visualization:

```
> plot(Clay1, Clay5, xlim=c(10,80), ylim=c(10,80), pch=as.numeric(zone),
+     xlab="Topsoil clay %", ylab="Subsoil clay %")
> title("Subsoil vs. topsoil clay, by zone")
> legend(10,75, pch=1:4, lty=1:4, legend=1:4)
> text(65, 40, "Slopes:")
> for (z in 1:4) {
+   m <- lm(Clay5 ~ Clay1, subset=(zone==z))
+   text(65, 40-(3*z), paste("zone",z,":",round(coefficients(m)[2], 3)))
+   m.l <- loess(Clay5 ~ Clay1, subset=(zone==z), span=100)
+   lines(y=c(min(m.l$fitted), max(m.l$fitted)),
+           x=c(min(m.l$x), max(m.l$x)), lty=z, lwd=1.5)
+ }
> m <- lm(Clay5 ~ Clay1)
> abline(m)
> text(65, 25, paste("overall:", round(coefficients(m)[2], 3)))
> rm(m, m.l, z)
```



Q82 : *Do the different regressions appear different? How different are the slopes? Referring back to the combined model summary, can we reject the null hypothesis that these slopes are in fact the same?* *Jump to A82* •

Q83 : *What are the reasons why an apparent difference that is readily visible is not statistically-significant?* *Jump to A83* •

7.9 Analysis of covariance*

In the parallel-lines model (§7.5) there was only one regression line between the continuous predictor and predictand, which could be moved up and down according to different class means; this is an *additive* model. In the interactions model (§7.8) there was both an overall line and deviations from it according to class, allowing different slopes, as well as differences in class means. Another way to look at this is to abandon the idea of a single regression altogether, and fit a separate line for each class. This is a *nested* model: the continuous predictor is measured only *within* each level of the classified predictor. It is specified with the `/` formula operator:

```
> lm51.z.n <- lm(Clay5 ~ zone/Clay1); summary(lm51.z.n)

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
zone1:Clay1   0.8343     0.1265    6.59  8.2e-10 ***
zone2:Clay1   0.7388     0.0625   11.83  < 2e-16 ***
zone3:Clay1   0.5640     0.0829    6.80  2.8e-10 ***
zone4:Clay1   1.0814     0.1387    7.80  1.3e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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
```

Note that there is no entry for `Clay1` by itself; rather there is a separate slope for each zone, e.g. `zone1:Clay1` for zone 1. The t-test is then whether each slope separately is different from 0.

Q84 : *How much of the variation in subsoil clay is explained by this model? How does this compare with the additive (parallel) model (§7.5) and the interactions model (§7.8)? Are all terms significant?* *Jump to A84* •

Q85 : *Compare the slopes for zones 1 and 4 in the nested model with the zone slopes (i.e. combined plus zone-specific) for these zones in the interaction model. Are they the same?* *Jump to A85* •

```
> coefficients(lm51.z.n)["zone4:Clay1"] -
+         (coefficients(lm51.z)["Clay1"] +
+         coefficients(lm51.z)["Clay1:zone4"])

zone4:Clay1
-1.1102e-15
```

ANCOVA

This model is also called the *Analysis of Covariance* (ANCOVA) when the aim is to detect differences in the classified predictor (here, zone), controlling for the effect of a continuous covariate, here the topsoil clay, when the covariate is considered a 'nuisance' parameter, not an object of the study.

In this case topsoil clay is not a nuisance parameter, but we can still see if controlling for it changes our perception of the differences between zones for subsoil clay.

Q86 : *Are the coefficients and significance levels between subsoil clay contents in the four zones different in the nested and additive models, and also the model which did not consider the covariate at all?* *Jump to A86* •

```
> summary(lm5z); summary(lm51.z.n); summary(lm51.z)

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

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

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
zone1:Clay1   0.8343     0.1265    6.59  8.2e-10 ***
zone2:Clay1   0.7388     0.0625   11.83  < 2e-16 ***
zone3:Clay1   0.5640     0.0829    6.80  2.8e-10 ***
zone4:Clay1   1.0814     0.1387    7.80  1.3e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

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

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 *
Clay1         0.8343     0.1265    6.59  8.2e-10 ***
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:zone2  -0.0955     0.1411   -0.68    0.500
Clay1:zone3  -0.2703     0.1513   -1.79    0.076 .
Clay1:zone4   0.2471     0.1877    1.32    0.190
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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
```

7.10 Design matrices for combined models*

In §6.3 we examined the *design matrix* for ANOVA and regression models. It is instructive to see this matrix for the combined models: additive (parallel), interactive, and nested. As in §6.3 we'll look at the matrix for observations 15 through 22, since they come from different zones.

```
> model.matrix(lm5z1)[15:22,]

   (Intercept) zone2 zone3 zone4 Clay1
15           1     0     0     0    22
16           1     1     0     0    52
17           1     1     0     0    20
18           1     1     0     0    33
19           1     0     1     0    21
20           1     0     1     0    22
21           1     0     1     0    26
22           1     0     0     1    38
```

```
> model.matrix(lm51.z)[15:22,]

   (Intercept) Clay1 zone2 zone3 zone4 Clay1:zone2 Clay1:zone3
15           1    22     0     0     0           0           0
16           1    52     1     0     0          52           0
17           1    20     1     0     0          20           0
18           1    33     1     0     0          33           0
19           1    21     0     1     0           0          21
20           1    22     0     1     0           0          22
21           1    26     0     1     0           0          26
22           1    38     0     0     1           0           0
   Clay1:zone4
15           0
16           0
17           0
18           0
19           0
20           0
21           0
22          38

> model.matrix(lm51.z.n)[15:22,]

   (Intercept) zone2 zone3 zone4 zone1:Clay1 zone2:Clay1 zone3:Clay1
15           1     0     0     0          22           0           0
16           1     1     0     0           0          52           0
17           1     1     0     0           0          20           0
18           1     1     0     0           0          33           0
19           1     0     1     0           0           0          21
20           1     0     1     0           0           0          22
21           1     0     1     0           0           0          26
22           1     0     0     1           0           0           0
   zone4:Clay1
15           0
16           0
17           0
18           0
19           0
20           0
21           0
22          38
```

Observation 15 is in zone 1, so it only has an entry for the intercept and topsoil clay. Observation 16 is in zone 2, so it has an entry for the intercept, topsoil clay, zone 2, and (in the second case) the interaction between topsoil clay and its zone. Note that for the parallel regression model `lm5z1` there is only one column for the continuous predictor, whereas in the interaction model `lm51.z` there is a separate column for the continuous predictor in each zone. This is how the model can fit a separate slope for each zone. In the nested model `lm51.z.n` there is no column for slope difference nor for overall slope, but rather one slope per zone, each with only the topsoil clay observations for that zone.

7.11 Answers

A59 : *CEC is positively correlated with both clay and organic matter; however there more spread in the CEC-vs-clay relation. The two possible predictors (clay and organic matter) are also positively correlated.* *Return to Q59* •

A60 : *The covariances depend on the measurement scales, whereas the correlations are standardised to the range* $[-1, 1]$. *CEC is highly correlated* ($r = 0.74$) *with organic carbon and somewhat less so* ($r = 0.56$) *with clay content. The two predictors are also moderately correlated* ($r = 0.60$). *Return to Q60* •

A61 : *These are given by the adjusted* R^2: *0.3066 using only clay as a predictor* ($CEC = 4.83 + 0.20 \cdot Clay$), *0.5489 using only organic carbon as a predictor* ($CEC = 3.67 + 2.52 \cdot OC$), *and 0.5662 using both together* ($CEC = 2.72 + 2.16 \cdot OC + 0.64 \cdot Clay$). *Return to Q61* •

A62 : *The predictive equation is only a little affected: the slope associated with OC decreases from 2.52 to 2.16, while the intercept (associated with no clay or organic carbon) decreases by 0.95. Adding Clay increases* R^2 *by only* $0.5662 - 0.5489 = 0.0173$, *i.e. 1.7%. This is significant* ($p = 0.010152$) *at the* $\alpha = 0.05$ *but not the* $\alpha = 0.01$ *level.* *Return to Q62* •

A63 : *OC only:* 0.549; *Clay only:* 0.307; *Both:* 0.566. *The model with both is slightly better than the single-predictor model from OC.* *Return to Q63* •

A64 : *The AIC favours the model with both OC and clay, but this is only slightly better than the single-predictor model from OC.* *Return to Q64* •

A65 : *The combined model has the lowest RSS (necessarily); the difference is only* 76.4, *i.e. about* 12% *lower. There is a* 1% *probability that this reduction is due to chance.* *Return to Q65* •

A66 : *The residuals are not normally-distributed; both tails are too long, and there are about six serious under-predictions (observations 73, 60, 63, 140, 77, 124).*

The two observations with the most negative residuals (over-predictions), i.e. 1 and 10, are the only two with very high clay and *OC*[5]. *This suggests an* interaction *at high levels; "the whole is more than the sum of the parts".*

There seems to be no comparable explanations for the four observations with the most positive residuals (under-predictions). *Return to Q66* •

A67 : *The model explains 50% (zone); 80% (topsoil clay); 82.5% (both) of the variation*

[5] `obs[(Clay1 > 60) & (OC1 > 5.5),]`

in subsoil clay; the combined model is only a bit better than the model using only measured topsoil clay. *Return to Q67* •

A68 : *The regression lines for zones 2, 3, and 4 are adjusted by* 5.69, 2.25, *and* −0.66, *respectively, compared to zone 1. These are the mean differences. The slope is* 0.736, *which is somewhat flatter than the slope estimated without considering zones,* 0.829. *That is, some of the apparently steep slope in the univariate model is accounted for by the differences between zones. In particular zone 2, which has the higher clay values in both layers, has a higher mean, so that once this is accounted for the regression line is not "pulled" to the higher values.* *Return to Q68* •

A69 : *Topsoil clay is very highly significant (*$p \approx 0$ *that it isn't) and so is the intercept (0 clay and zone 1). Zone 2 is significantly different (*$p < 0.008$ *that it isn't) but the others are not. Note that in the one-way ANOVA by zone, zones 3 and 4 are both significantly different from zone 1 and 2, which form a group. Here we see that the inclusion of topsoil clay in the model has completely changed the relation to zone, since much of the zone effect was in fact a clay effect, i.e. zones had different average topsoil clay contents. The two predictors were* confounded. *Return to Q69* •

A70 : *The residuals are more or less normally distributed around 0, except for one very large negative residual (under-prediction) and seven large positive residuals (heavy tail)* *Return to Q70* •

A71 : *At point 145, the prediction is 23% while the actual is 47%; this is a severe under-prediction. This is an unusual observation: topsoil clay is 7% higher than both underlying layers. There are only two observations where topsoil clay exceeds subsoil clay (*`> which(Clay1 > Clay5)`*), 145 and 81, and for observation 81 the difference is only 2%.*

At point 119, the prediction is 34% while the actual is 48%; this is the largest under-prediction. Here topsoil clay is fairly low (21%) compared to the much higher subsoil values. *Return to Q71* •

A72 : *Variables* `Clay1` *and* `Clay2` *have* $VIF \geq 10$ *and are thus highly co-linear with other variables. As a set, the others are fairly independent.* *Return to Q72* •

A73 : *If either* `Clay1` *or* `Clay2` *are removed, the remaining set of five variables are fairly independent (all* $VIF < 5$*). This shows that the high VIF for* `Clay1` *and* `Clay2` *in the full model was due to the presence of the other "clay" variable. So either topsoil or subsoil clay should be included in a parsimonious model, but not both.* *Return to Q73* •

A74 : *Eliminating* `Clay1` *results in a much lower AIC. This seems logical, as subsoil clay (*`Clay2`*) is closer physically to the deep subsoil (target variable* `Clay5`*), so the processes that lead to a certain clay content would seem to be more similar.* *Return to Q74* •

A75 : *The final stepwise regression model, starting from the full set less* `Clay1`*, is*

`Clay5 ~ Clay2 + CEC2`, *with an AIC of 835.2. The model starting from the full set is* `Clay5 ~ Clay1 + Clay2 + CEC2`, *i.e. it has both clays as well as the subsoil CEC. Its AIC is 834.94. The two final models are almost the same except for the inclusion of the highly-colinear variable; their AIC is almost identical. So, the reduced model (without* `Clay1` *is preferred.* *Return to Q75* •

A76 : *Both* `Clay1` *and* `Clay2` *have* VIF > 8, *not above the threshold* VIF >= 10 *but not much below. Clearly,* `Clay1` *and* `Clay2` *are still highly-correlated.* *Return to Q76* •

A77 : *As in the previous tasks of this section, we see that* `Clay1` *can be eliminated with almost no increase in model information content as shown by the AIC.*

Return to Q77 •

A78 : *Zone 4 (blue points and line, low clay values) seems poorly-fit. A line with a lower intercept and a steeper slope would appear to fit better. So a model with interaction between classified and continuous predictor, allowing separate slopes for each class, might be better. For the other three the parallel lines seem OK.* *Return to Q78* •

A79 : *The model explains 83.4% of the variation in subsoil clay; this is slightly better than the additive model (82.5%).* *Return to Q79* •

A80 : *Additive terms for topsoil clay, the intercept (zone 1 at zero clay) and zone 3 are significant. This differs from the additive model, where zone 2 was the only zone significantly different from the intercept.* *Return to Q80* •

A81 : *The most significant interaction is* `Clay1:zone3` *but the probability that rejecting the null hypothesis of no difference in slopes is fairly high,* 0.076, *so we can't reject the null hypothesis at the conventional 95% confidence level.* *Return to Q81* •

A82 : *They certainly* appear *different, ranging from* 0.564 *in zone 3 (green points and line) to 1.081 (blue points and line), almost double. Yet the* t*-tests for the interaction terms are not significant at the 95% confidence level, so these four slopes could all be different just because of sampling error.* *Return to Q82* •

A83 : *The fundamental problems are: (1) small sample size in each zone; (2) a spread of points ("cloud" or "noise") within each zone. These two factors make it difficult to establish statistical significance.* *Return to Q83* •

A84 : *The nested model explains 83.4% of the variation in subsoil clay; this is slightly better than the additive model (82.5%) and the same as the interactions model. It is quite unlikely that the mean for zone 4 is different from zone 1.* *Return to Q84* •

A85 : *Yes, they are the same. For zone 1, the interaction model has the default slope (coefficient for* `Clay1`*) which is the same as the nested model slope for zone 1*

(coefficient for `zone1:Clay1`*). For zone 4, adding the slope difference in the interaction model (coefficient for* `Clay1:zone4`*) to the default slope (coefficient for* `Clay1`*) gives the same value as the nested model slope for zone 4 (coefficient for* `zone4:Clay1`*).* *Return to Q85* •

A86 : *There is a big difference between the model coefficients and their significance. Without considering the covariate at all, the difference from zone 1 is (zone 4 ≫ zone 3 ≫ zone 2), the latter is not significantly different. In the nested model the differences are (zone 3 > zone 2 ≫ zone 4), the latter coefficient not significant; this is because the difference between zone 1 and 4 subsoil clay can be almost entirely explained if one knows the topsoil clay and allows separate regression lines for each zone. In the additive (parallel) model the differences are (zone 2 > zone 3 ≫ zone 4). The parallel regression line for zone 2 is significantly above that for zone 1, the others not significantly different.* *Return to Q86* •

8 Factor analysis

Sometimes we are interested in the inter-relations between a set of variables, not just their individual (partial) correlations (§7.1). That is, we want to investigate the *structure* of the *multivariate feature space* covered by a set of variables.; this is *factor analysis*. This can also be used to diagnose multi-collinearity and select representative variables (see also §7.6).

The basic idea is that the *vector space* made up of the original variables may be *projected* onto another space, where the new *synthetic variables* are *orthogonal* to each other, i.e. completely uncorrelated. These synthetic variables can often be *interpreted* by the analyst, that is, they represent some composite attribute of the objects of study.

8.1 Principal components analysis

The first such technique is *Principal components analysis*. This is a multivariate data reduction technique. It finds a new set of variables, equal in number to the original set, where these *synthetic variables* are uncorrelated (i.e. orthogonal to each other in the space formed by the principal components). In addition, the first synthetic variable represents as much of the common variation of the original variables as possible, the second variable represents as much of the residual variation as possible, and so forth.

> **Note:** This is a common image-processing technique and is explained and illustrated in many textbooks on remote sensing [e.g. 1, 20].

In the present example, we investigate the structure of the feature space defined by the three variables (CEC, Clay, and OC) in a single horizon. A summary of the components reveals how much redundancy there is in this space.

Task 58 : Compute the unstandardized principal components of three variables: topsoil clay, CEC, and organic carbon. •

To compute the PCs we use the `prcomp` method; this produces an object of class `prcomp` which contains information about the components. The relevant columns are extracted from the data frame.

```
> pc <- prcomp(obs[,c("CEC1","Clay1","OC1")])
> class(pc)

[1] "prcomp"

> str(pc)

List of 5
 $ sdev    : num [1:3] 14.282 4.192 0.933
 $ rotation: num [1:3, 1:3] -0.2187 -0.9735 -0.0665 -0.9589 0.2271 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "CEC1" "Clay1" "OC1"
  .. ..$ : chr [1:3] "PC1" "PC2" "PC3"
 $ center  : Named num [1:3] 11.2 31.27 2.99
  ..- attr(*, "names")= chr [1:3] "CEC1" "Clay1" "OC1"
 $ scale   : logi FALSE
 $ x       : num [1:147, 1:3] -40.3 -39 -31.5 -23.2 -16.2 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:147] "1" "2" "3" "4" ...
  .. ..$ : chr [1:3] "PC1" "PC2" "PC3"
 - attr(*, "class")= chr "prcomp"

> summary(pc)

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
```

Q87 : *What proportion of the total variance is explained by the first component alone? By the first and second?* *Jump to A87* •

The numbers here are misleading, because the variables are on different scales. In these cases it is better to compute the *standardised* components, using the correlation instead of covariance matrix; this standardises all the variables to zero mean and unit standard deviation before computing the components.

Task 59 : Compute the standardized principal components of three variables: topsoil clay, CEC, and organic carbon. •

This option is specified by setting the `scale` optional argument to `TRUE`.

```
> pc.s <- prcomp(obs[c(10,7,13)], scale=T)
> summary(pc.s)

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
```

Q88 : *What is the difference between the variance proportions in the standardized vs. unstandardized principal components? Which gives a better idea of the proportion of variance explained? In what circumstances would you prefer to use unstandardized components?* *Jump to A88* •

Q89 : *What proportion of the total standardised variance is explained by the first component alone? By the first and second?* *Jump to A89* •

We can see which original variables are associated with which synthetic variables by examining the *loadings*, also called the factor *rotations*. These are the eigenvectors (in the columns) which multiply the original variables to produce the synthetic variables (principal components).

```
> 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
```

These show the amount that each original (standardised) original variable contributes to each synthetic variable. Here, the first PC is an almost equal mixture of CEC, Clay, and OC; this can be interpreted as an **overall intensity of soil activity**; we've seen that CEC, Clay and OC are generally all positively-correlated and this strong relation comes out in the first PC. This represents 76% of the overall variability. The second PC has a large contribution from Clay opposed to the two other variables; this component can be interpreted as high CEC without high Clay, i.e. **high CEC due mostly to OC**. This represents 16% of the overall variability. The third PC represents CEC that is higher than expected by the OC content. The interpretation here is more difficult. It could just represent lack of precision in the laboratory (i.e. experimental error). Or it could represent a different compostion of the organic matter. This represents 8% of the overall variability.

8.1.1 The synthetic variables*

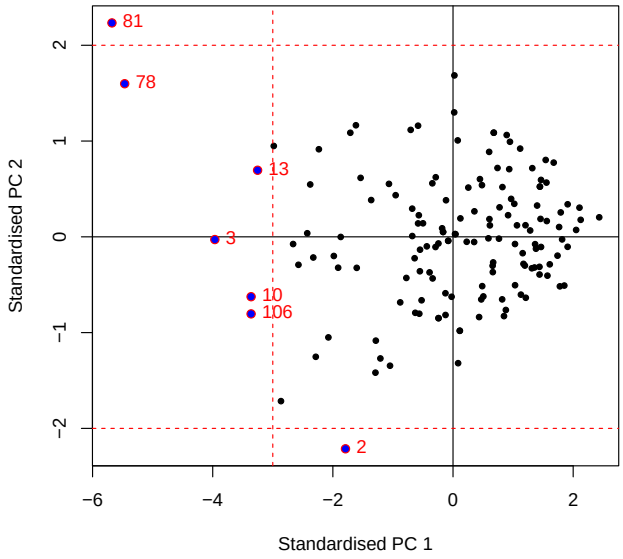
If the `retx` argument to the `prcomp` method is specified as `TRUE` (this is the default), R computes the numeric value of each observation for each PC; these are the *scores* and are the values of the new variables, substituting for the original variables. They are stored in the `x` field of the `prcomp` object.

It's instructive to see what the observations look like in the space spanned by the PCs. (These are also displayed as part of the biplot, see §8.1.3.)

Task 60 : Compute the standardized PCs, along with the scores for each obser-

vation. Plot these in the space spanned by the first two PCs and highlight the observations that are not well-explained by these. •

```
> pc.s <- prcomp(obs[c(10,7,13)], scale=T, retx=T)
> plot(pc.s$x[,1], pc.s$x[,2], pch=20,
+    xlab="Standardised PC 1", ylab="Standardised PC 2")
> abline(h=0); abline(v=0)
> abline(h=2, col="red", lty=2); abline(v=3, col="red", lty=2)
> abline(h=-2, col="red", lty=2); abline(v=-3, col="red", lty=2)
> (pts <- which((abs(pc.s$x[,1]) >= 3) | (abs(pc.s$x[,2]) >= 2) ))

  2   3  10  13  78  81 106
  2   3  10  13  78  81 106

> points(pc.s$x[pts,1], pc.s$x[pts,2], pch=21, col="red", bg="blue")
> text(pc.s$x[pts,1], pc.s$x[pts,2], pts, pos=4, col="red")
> pc.s$x[pts, c(1,2)]

        PC1       PC2
2   -1.7900 -2.213046
3   -3.9647 -0.028197
10  -3.3611 -0.625042
13  -3.2530  0.695328
78  -5.4654  1.598616
81  -5.6773  2.233634
106 -3.3612 -0.804394

> obs[pts, c(10,7,13)]

    CEC1 Clay1   OC1
2   12.6    71  3.20
3   21.7    61  6.98
10  14.9    62  7.34
13  23.3    48  6.00
78  29.0    53  9.40
81  28.0    46 10.90
106 22.0    67  4.80
```



In the displayed graph, we can identify unusual points (towards the outside of the plot) with the `identify` method and then display their values of the original variables. In this example, points furthest from the centroid are indentified by their distance and plotted in a different colour; the `identify` method wasn't used here because it is interactive.

Q90 : *Which are the most unusual observations in the space spanned by the first two PC's? What do these represent, in terms of the original variables?* *Jump to A90* •

8.1.2 Residuals*

The PC scores, along with the loadings (rotations), contain the complete information of the original observations.

Task 61 : Confirm this: reproduce the original observations from the results of the PCA and compare to the original observations. •

By default, the `prcomp` function *centres* each variable on zero (by subtracting the mean), but does not by defaults *scale* them (by dividing by the standard deviation). This is done simply to avoid problems with the numeric solution. It's easier to see that the multiplication of the score matrix by the rotation matrix reproduces the original values with the non-centred and non-scaled PCA. So, specify argument `center` to be `FALSE`.

First, we compute the PCs without any centring or scaling, and confirm that the rotations are used to produce the synthetic variables:

```
> pc <- prcomp(obs[,c("CEC1","Clay1","OC1")], retx=TRUE, center=FALSE)
> summary(as.matrix(obs[,c("CEC1","Clay1","OC1")])%*%pc$rotation - pc$x)
```

```
      PC1          PC2          PC3
 Min.   :0   Min.   :0   Min.   :0
 1st Qu.:0   1st Qu.:0   1st Qu.:0
 Median :0   Median :0   Median :0
 Mean   :0   Mean   :0   Mean   :0
 3rd Qu.:0   3rd Qu.:0   3rd Qu.:0
 Max.   :0   Max.   :0   Max.   :0
```

The observations multiplied by eigenvectors indeed are the synthetic variables.

Now we invert the process, to reproduce the orginal values from the synthetic ones. Since we have the relation:

$$OE = S \tag{7}$$

where O are the original observations (147×3), E are the eigenvectors (3×3), and X are the values of the synthetic variables (147×3), we then must have:

$$O = SE^{-1} \tag{8}$$

We find the inverse of the rotations matrix with the `solve` function with only one argument (the matrix to be inverted); this then post-multiplies the scores matrix:

```
> obs.reconstruct <- pc$x %*% solve(pc$rotation)
> summary(obs.reconstruct - obs[,c("CEC1","Clay1","OC1")])
```

```
      CEC1                Clay1                 OC1
 Min.   :-7.11e-15   Min.   :-1.42e-14   Min.   :-1.78e-15
 1st Qu.:-1.78e-15   1st Qu.:-1.78e-15   1st Qu.:-4.44e-16
 Median :-1.78e-15   Median : 0.00e+00   Median : 0.00e+00
 Mean   :-1.36e-15   Mean   :-8.70e-16   Mean   :-1.59e-16
 3rd Qu.: 0.00e+00   3rd Qu.: 0.00e+00   3rd Qu.: 0.00e+00
 Max.   : 1.78e-15   Max.   : 1.42e-14   Max.   : 8.88e-16
```

The only difference between the reconstructed observations and the originals is due to limited computational precision; mathematically they are identical

If fewer than the maximum PCs are used, they will not exactly reproduce the original observations. By omitting the higher PCs, we are sacrificing some information for increased parsimony. The question is, how much?

Task 62 : Compute the orginal value of the observations, using only the first standardized PC, and then with the first two. Compute the residuals. •

Here we use only the first eigenvector, and then the first two. In both cases we have to use only the first scores and the first rows of the inverted rotation matrix. Note the use of the `drop` argument when selecting only one row or column of a matrix with the `[` "extract" method. By default this is `TRUE` (and invisible); any extra dimensions are dropped, and so selecting only one row or column results in vector, not a matrix, and so can not be used in matrix operations. When this is `FALSE` the dimensions are retained.

For completeness, we show the long form of the full reconstruction. Note that residuals are defined as (observed - modelled).

```
> dim(solve(pc$rotation)[1,])

NULL

> dim(solve(pc$rotation)[1,,drop=T])

NULL

> dim(solve(pc$rotation)[1,,drop=F])

[1] 1 3

> obs.reconstruct.1 <- pc$x[,1,drop=F] %*% solve(pc$rotation)[1,,drop=F]
> summary(obs[,c("CEC1","Clay1","OC1")] - obs.reconstruct.1)

      CEC1              Clay1              OC1
 Min.   :-10.187   Min.   :-4.505   Min.   :-2.9565
 1st Qu.: -2.303   1st Qu.:-0.936   1st Qu.:-0.5964
 Median :  0.203   Median :-0.129   Median : 0.0123
 Mean   :  0.546   Mean   :-0.195   Mean   : 0.1079
 3rd Qu.:  2.548   3rd Qu.: 0.800   3rd Qu.: 0.7975
 Max.   : 13.071   Max.   : 3.721   Max.   : 6.2981

> obs.reconstruct.2 <- pc$x[,1:2] %*% solve(pc$rotation)[1:2,]
> summary(obs[,c("CEC1","Clay1","OC1")] - obs.reconstruct.2)

      CEC1               Clay1                OC1
 Min.   :-0.78864   Min.   :-0.118803   Min.   :-3.2491
 1st Qu.:-0.09594   1st Qu.:-0.014452   1st Qu.:-0.5447
 Median : 0.00821   Median : 0.001237   Median :-0.0439
 Mean   :-0.00201   Mean   :-0.000304   Mean   : 0.0108
 3rd Qu.: 0.10186   3rd Qu.: 0.015344   3rd Qu.: 0.5130
 Max.   : 0.60759   Max.   : 0.091529   Max.   : 4.2173

> obs.reconstruct <- pc$x[,1:3] %*% solve(pc$rotation)[1:3,]
> summary(obs[,c("CEC1","Clay1","OC1")] - obs.reconstruct)

      CEC1                Clay1                 OC1
 Min.   :-1.78e-15   Min.   :-1.42e-14   Min.   :-8.88e-16
 1st Qu.: 0.00e+00   1st Qu.: 0.00e+00   1st Qu.: 0.00e+00
 Median : 1.78e-15   Median : 0.00e+00   Median : 0.00e+00
 Mean   : 1.36e-15   Mean   : 8.70e-16   Mean   : 1.59e-16
 3rd Qu.: 1.78e-15   3rd Qu.: 1.78e-15   3rd Qu.: 4.44e-16
 Max.   : 7.11e-15   Max.   : 1.42e-14   Max.   : 1.78e-15
```

Q91 : *What happens to the accuracy of the reconstruction as the number of components is increased?* *Jump to A91* •

Task 63 : Create a matrix of the residuals that result from using only one and two PCs to represent the three variables. •

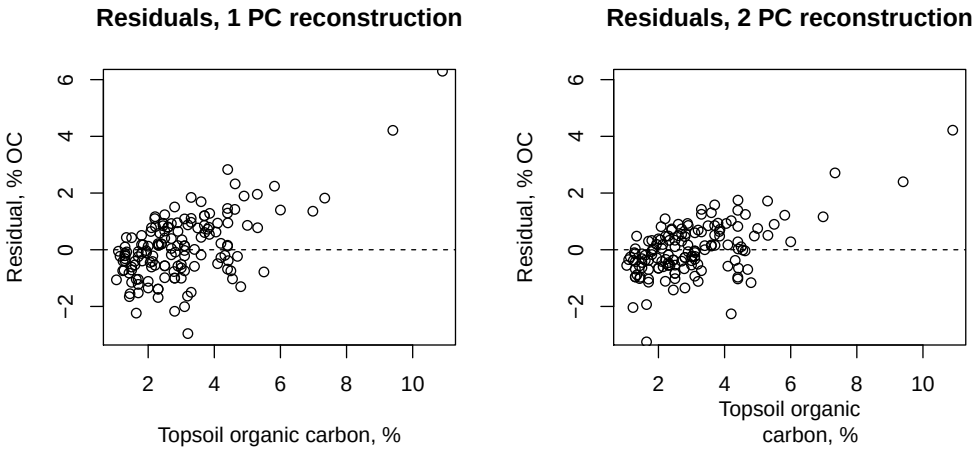
The previous code has done this, but not returned it as a separate object.

```
> resid.reconstruct.1 <- obs[,c("CEC1","Clay1","OC1")] -
+                                    pc$x[,1,drop=F] %*% solve(pc$rotation)[1,,drop=
> summary(resid.reconstruct.1)

      CEC1              Clay1              OC1
 Min.   :-10.187   Min.   :-4.505   Min.   :-2.9565
 1st Qu.: -2.303   1st Qu.:-0.936   1st Qu.:-0.5964
 Median :  0.203   Median :-0.129   Median : 0.0123
 Mean   :  0.546   Mean   :-0.195   Mean   : 0.1079
 3rd Qu.:  2.548   3rd Qu.: 0.800   3rd Qu.: 0.7975
 Max.   : 13.071   Max.   : 3.721   Max.   : 6.2981

> head(sort(resid.reconstruct.1[,"OC1"]))

[1] -2.9565 -2.2345 -2.1700 -2.0076 -1.6831 -1.6532

> head(sort(resid.reconstruct.1[,"OC1"], decreasing=T))

[1] 6.2981 4.2113 2.8299 2.3241 2.2431 1.9575

> resid.reconstruct.2 <- obs[,c("CEC1","Clay1","OC1")] -
+                                    pc$x[,1:2] %*% solve(pc$rotation)[1:2,]
> summary(resid.reconstruct.2)

      CEC1               Clay1                OC1
 Min.   :-0.78864   Min.   :-0.118803   Min.   :-3.2491
 1st Qu.:-0.09594   1st Qu.:-0.014452   1st Qu.:-0.5447
 Median : 0.00821   Median : 0.001237   Median :-0.0439
 Mean   :-0.00201   Mean   :-0.000304   Mean   : 0.0108
 3rd Qu.: 0.10186   3rd Qu.: 0.015344   3rd Qu.: 0.5130
 Max.   : 0.60759   Max.   : 0.091529   Max.   : 4.2173

> head(sort(resid.reconstruct.2[,"OC1"]))

[1] -3.2491 -2.2603 -2.0374 -1.9332 -1.4185 -1.3474

> head(sort(resid.reconstruct.2[,"OC1"], decreasing=T))

[1] 4.2173 2.7102 2.3981 1.7506 1.7186 1.5830
```

Task 64 : Plot the residuals vs. original values of organic carbon for the one- and two-PC cases, using the same scale. •

```
> par(mfrow=c(1,2))
> ymax <- round(max(resid.reconstruct.1[,"OC1"],
+                                  resid.reconstruct.2[,"OC1"]))
> ymin <- round(min(resid.reconstruct.1[,"OC1"],
+                                 resid.reconstruct.2[,"OC1"]))
> plot(resid.reconstruct.1[,"OC1"] ~ obs[,"OC1"],
+        main="Residuals, 1 PC reconstruction",
+        xlab="Topsoil organic carbon, %",
+        ylab="Residual, % OC", ylim=c(ymin,ymax))
> abline(h=0, lty=2)
> plot(resid.reconstruct.2[,"OC1"] ~ obs[,"OC1"],
+        main="Residuals, 2 PC reconstruction", xlab="Topsoil organic
+        carbon, %",
```

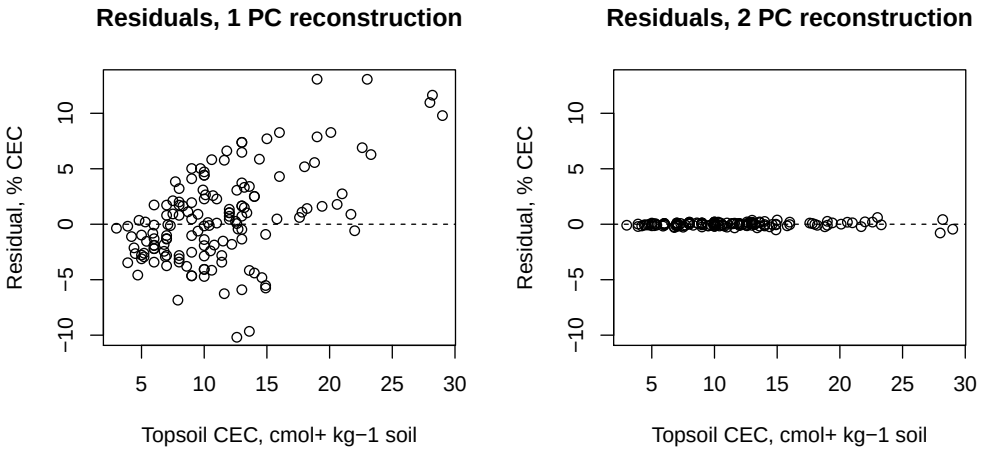
```
+        ylab="Residual, % OC", ylim=c(ymin,ymax))
> abline(h=0, lty=2)
> par(mfrow=c(1,1))
```



Q92 : *What is the pattern of the reconstruction residuals? Try to explain. (Hint: look at the loadings,* `pc$rotation`*.) Are two PCs satisfactory for representing topsoil carbon?* *Jump to A92* •

Task 65 : Repeat the analysis for CEC. •

```
> par(mfrow=c(1,2))
> ymax <- round(max(resid.reconstruct.1[,"CEC1"],
+                           resid.reconstruct.2[,"CEC1"]))
> ymin <- round(min(resid.reconstruct.1[,"CEC1"],
+                           resid.reconstruct.2[,"CEC1"]))
> plot(resid.reconstruct.1[,"CEC1"] ~ obs[,"CEC1"],
+              main="Residuals, 1 PC reconstruction",
+              xlab="Topsoil CEC, cmol+ kg-1 soil",
+              ylab="Residual, % CEC", ylim=c(ymin,ymax))
> abline(h=0, lty=2)
> plot(resid.reconstruct.2[,"CEC1"] ~ obs[,"CEC1"],
+              main="Residuals, 2 PC reconstruction",
+              xlab="Topsoil CEC, cmol+ kg-1 soil",
+              ylab="Residual, % CEC", ylim=c(ymin,ymax))
> abline(h=0, lty=2)
> par(mfrow=c(1,1))
```



Q93 : *What is the pattern of the reconstruction residuals? Are two PCs satisfactory for representing topsoil CEC?* *Jump to A93*

•

It may help to answer this question if you compute the range of residuals in the two cases using the `range` function:

```
> range(resid.reconstruct.1[,"CEC1"])

[1] -10.187  13.071

> range(resid.reconstruct.2[,"CEC1"])

[1] -0.78864  0.60759
```

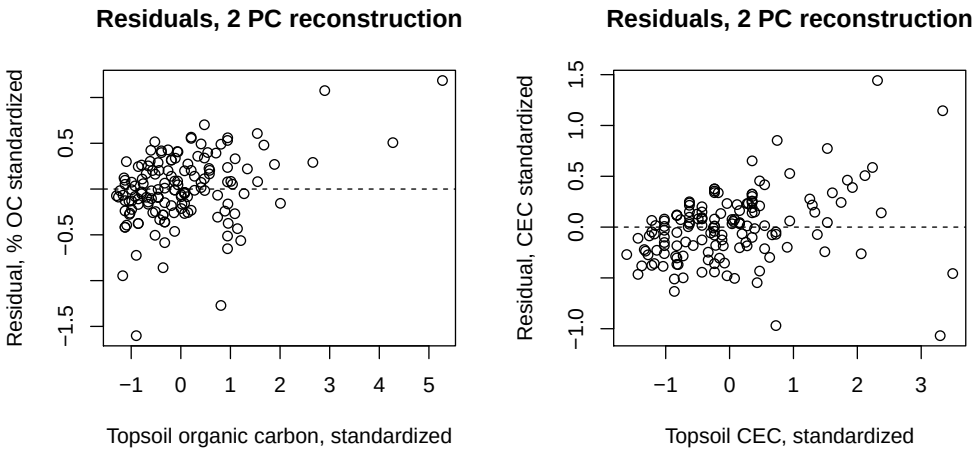
Standardized residuals can also be computed. The relation $O = SE^{-1}$ is *not* valid here, because the E matrix refers to eigenvectors from the **standardized** variables. However, we can standardize the original variables ourselves and then use these to compute residuals; i.e. S becomes standardized S_s and the same back-transformation can be used.

Task 66 : Compute the standarized residuals of organic carbon and CEC, for the two-PC case. Plot the residuals vs. original standardized values. •

We use the `scale` function to scale the columns of a matrix:

```
> resid.reconstruct.2 <- scale(obs[,c("CEC1","Clay1","OC1")]) -
+                                          pc.s$x[,1:2] %*% solve(pc.s$rotation)
> summary(resid.reconstruct.2)

      CEC1              Clay1               OC1
 Min.   :-1.068   Min.   :-0.1512   Min.   :-1.6011
 1st Qu.:-0.208   1st Qu.:-0.0294   1st Qu.:-0.1966
 Median : 0.012   Median : 0.0017   Median :-0.0133
 Mean   : 0.000   Mean   : 0.0000   Mean   : 0.0000
```

```
 3rd Qu.: 0.177   3rd Qu.: 0.0251   3rd Qu.: 0.2305
 Max.   : 1.443   Max.   : 0.2042   Max.   : 1.1855

> par(mfrow=c(1,2))
> plot(resid.reconstruct.2[,"OC1"] ~ scale(obs[,"OC1"]),
+        main="Residuals, 2 PC reconstruction",
+        xlab="Topsoil organic carbon, standardized",
+        ylab="Residual, % OC standardized")
> abline(h=0, lty=2)
> plot(resid.reconstruct.2[,"CEC1"] ~ scale(obs[,"CEC1"]),
+        main="Residuals, 2 PC reconstruction",
+        xlab="Topsoil CEC, standardized",
+        ylab="Residual, CEC standardized")
> abline(h=0, lty=2)
> par(mfrow=c(1,1))
```



Q94 : *How do these reconstructions compare to those using unstandardized PCs?* *Jump to A94* •

8.1.3 Biplots*

The relation between variables and observations in the new space can be visualised with a *biplot* [15, 17]. This has two kinds of information on one plot, leading to four interpretations:

1. The plot shows the *observations* as *points*, labeled by the observation number (i.e. the row in data frame), in the plane formed by two principal components (synthetic variables). Any two PC's can be used; most common are the first two.

   The coordinates of the points are shown in the lower (PC1) and left (PC2) margins; they are the transformed variables, with the origin $(0, 0)$ defined by the mean of the data in this space, and scaled to have similar ranges (this can be changed by an optional parameter).

These points are interepreted like any scatterplot: you can find clusters of observations or outliers in this space. Note that the scaling can affect your visualisation.

2. The plot shows the original *variables* from which the PC's were computed (i.e. the original feature space) as *vectors*. They begin at the origin and extend to coordinates shown by the upper (PC1) and right (PC2) margins. These have a *different* scale (and scaling) than the observations, so they must be interepreted separately.

   These can be interpreted in three ways:

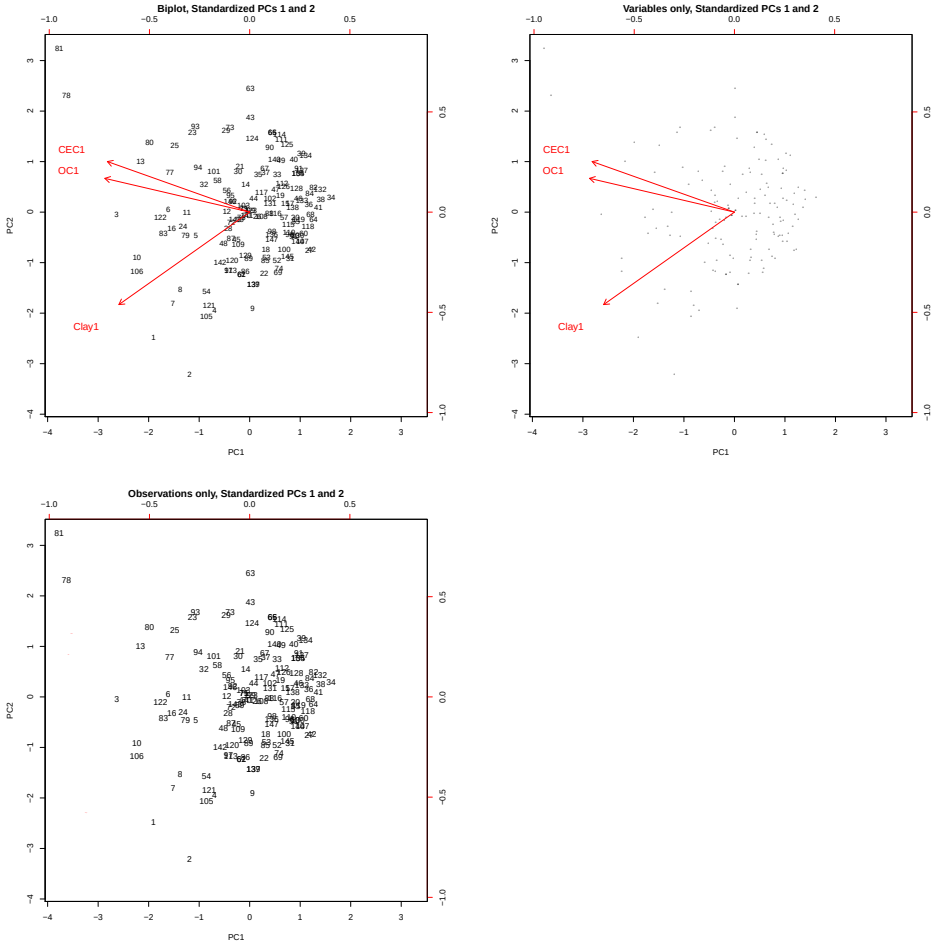
   (a) The *orientation* (direction) of the vector, with respect to the PC space, in particular, its angle with the PC axes: the *more parallel to a PC axis* is a vector, the more it contributes only to that PC. The contribution of an original variable to a PC can be estimated from the projection of the vector onto the PC.

   (b) The *length* in the space defined by the displayed PCs; the *longer* the vector, the more variability of this variable is represented by the two displayed PCs; short vectors are thus better represented in other dimensions (i.e. they have a component that is orthogonal to the plane formed by the two displayed PCs, which you can visualize as coming out of the plane).

   (c) The *angles between vectors* of different variables show their correlation in this space: small angles represent high positive correlation, right angles represent lack of correlation, opposite angles represent high negative correlation.

We can also produce versions of the biplot emphasizing only the vectors or only the points. These require some tricks with the arguments to the `biplot` method. In any event we exaggerate a bit the text size of the variables with the `cex=` argument.

```
> par(mfrow=c(2,2))
> biplot(pc.s, main="Biplot, Standardized PCs 1 and 2",
+        pc.biplot=T, cex=c(.9,1.2))
> biplot(pc.s, main="Variables only, Standardized PCs 1 and 2",
+        pc.biplot=T, cex=c(0.3,1.2), xlabs=rep("o", dim(pc.s$x)[1]))
> biplot(pc.s, main="Observations only, Standardized PCs 1 and 2",
+        pc.biplot=T, var.axes=F, cex=c(1,0.1))
> par(mfrow=c(1,1))
```



The argument `pc.biplot=T` produces a so-called "principal component biplot", where the observations are scaled up by $\sqrt{n}$ and variables scaled down by the same factor. With this scaling, inner products between variables (as shown by the vectors) approximate their correlations and distances between observations (as shown by the points) approximate Mahalanobis distance in PC space. These lead to easier visual interpretation.)

First, we can look for groups of points ("clusters") and unusual points ("outliers") in the new space.

Q95 : *Do there appear to be any clusters formed by the observations? If so, where?* *Jump to A95* •

Q96 : *Which* observations *are unusual in the space spanned by the first two standardized principal components? Can you explain them from the original observations?* *Jump to A96* •

```
> obs[c(1,2,78,81),c(7,10,13)]
```

```
   Clay1 CEC1  OC1
1     72 13.6  5.5
2     71 12.6  3.2
78    53 29.0  9.4
81    46 28.0 10.9
```

Second, we can look at the *vectors* representing the original variables.

Q97 : *Which variables are better explained in this space? (Hint: look at the length of the vectors.)* *Jump to A97* •

Q98 : *Which variables contribute most to PC1? to PC2? (Hint: look at the projection of the vectors onto the axes.)* *Jump to A98* •

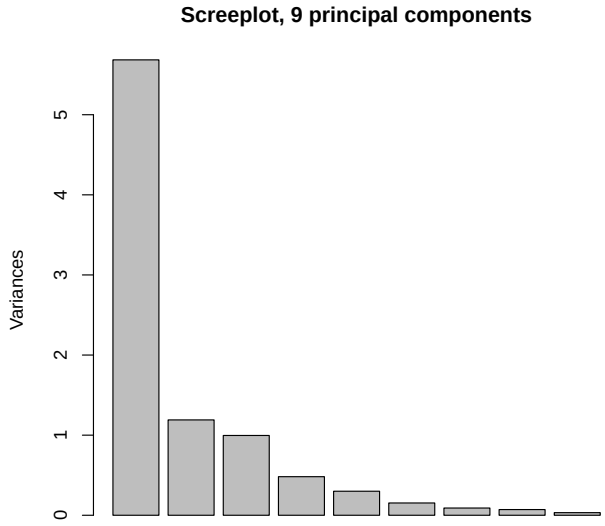
Q99 : *Which variables are highly-correlated in this space? (Hint: look at the angles between the vectors.) What does this imply for modelling?* *Jump to A99* •

8.1.4 Screeplots*

A useful graphical representation of the proportion of the total variance explained by each component is the *screeplot*. This is named for a "scree slope", which is the zone at the base of a steep cliff where debris ("scree") accumulates. We look for the "breaks" in the slope to decide how many PCs can be meaningfully interpreted.

Task 67 : Repeat this analysis, but with the three continuous variables from all three layers, i.e. a total of nine variables. Show the proportional variance explained with a screeplot. •

```
> pc9 <- prcomp(obs[7:15], scale=T)
> summary(pc9)

Importance of components:
                          PC1   PC2   PC3    PC4    PC5   PC6   PC7
Standard deviation     2.384 1.091 0.998 0.6939 0.5478 0.392 0.301
Proportion of Variance 0.632 0.132 0.111 0.0535 0.0333 0.017 0.010
Cumulative Proportion  0.632 0.764 0.875 0.9281 0.9614 0.978 0.988
                          PC8     PC9
Standard deviation     0.26680 0.18038
Proportion of Variance 0.00791 0.00362
Cumulative Proportion  0.99638 1.00000

> screeplot(pc9, main="Screeplot, 9 principal components")
```

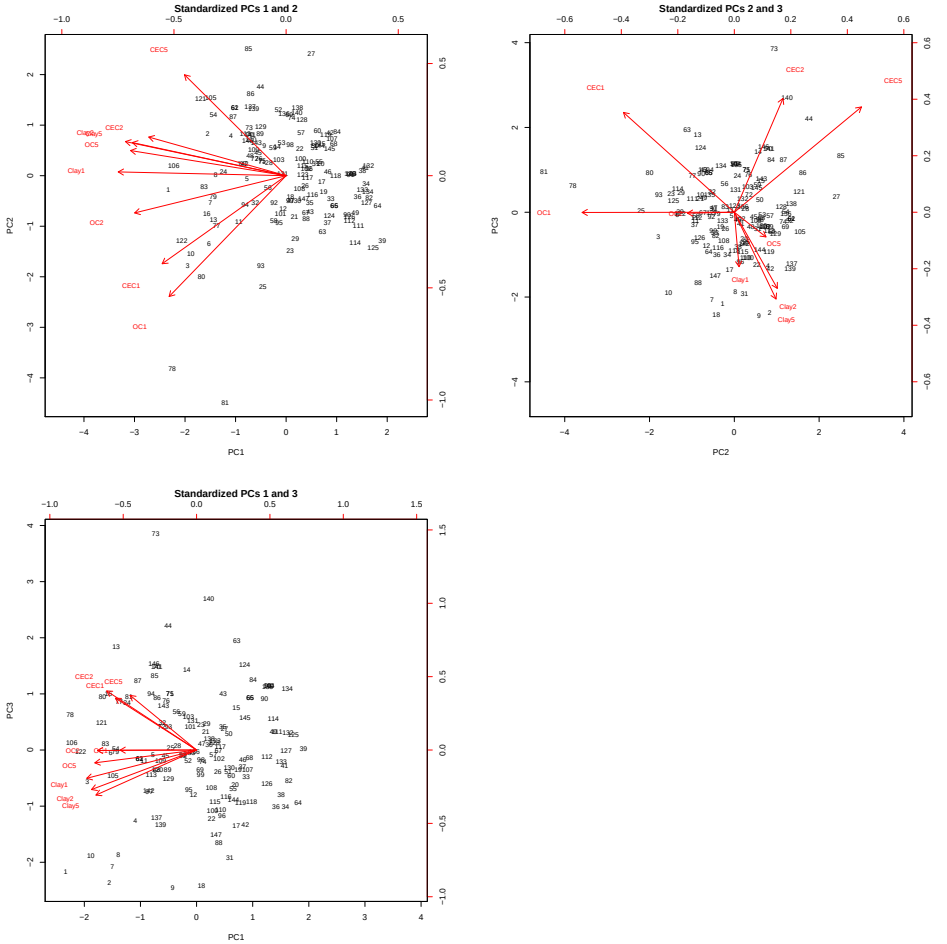


Q100 : *How much of the variance is explained by the first PC? At what components does the "scree slope" change substantially? How many PCs out of the 9 computed are meaningful?* *Jump to A100* •

Task 68 : Show the three faces of the cube defined by PCs 1, 2 and 3 as three 2-D biplots. •

To show combinations of PCs other than the first two, we must use the `choice=` argument.

```
> par(mfrow=c(2,2))
> biplot(pc9, pc.biplot=T, main="Standardized PCs 1 and 2")
> biplot(pc9, choice=2:3, pc.biplot=T, main="Standardized PCs 2 and 3")
> biplot(pc9, choice=c(1,3), pc.biplot=T, main="Standardized PCs 1 and 3")
> par(mfrow=c(1,1))
```



8.2 Factor analysis*

PCA is a data reduction technique, but the resulting synthetic variables may not be interpretable. A related technique is *factor analysis* in the sense used in social sciences [35, §11.3]. Here we hypothesize that the set of *observed variables* is a measureable expression of some (smaller) number of *latent variables* that can't themselves be measured, but which influence a number of the observed variables. This has an obvious interpretation in pyschology, where concepts such as "math ability" or "ability to think abstractly" can't be directly measured; instead these variables are assumed to exist (based on external evidence) and measured with various clever tests.

In the natural sciences the concept of latent variables is not so easy to justify; still, the techniques are useful because they (1) give a method to rotate axes to line up observed with synthetic variables and (2) allow us to determine how many latent variables there might be.

Suppose there are k original variables, to be explained by $p < k$ factors. Factor analysis decomposes the $k \times k$ variance-covariance matrix Σ of the original variables into a $p \times k$ loadings matrix Λ (the k columns are the original variables, the p rows are the factors) and a $k \times k$ diagonal matrix of unexplained variances

per original variable (its *uniqueness*) Ψ, such that

$$\Sigma = \Lambda'\Lambda + \Psi$$

In PCA $p = k$, there is no Ψ, and all variance is explained by the synthetic variables; there is only one way to do this. In factor analysis, the loadings matrix Λ is not unique; it can be multiplied by any $k \times k$ orthogonal matrix, known as *rotations*. The factor analysis algorithm finds a rotation to satisfy user-specified conditions; one common condition is known as *varimax*; this is the default in R.

Task 69 : Compute a factor analysis assuming three latent variable, over the three continuous variables from all three layers (i.e. nine original variables). •

```
> (fa <- factanal(obs[7:15], 3))

Call:
factanal(x = obs[7:15], factors = 3)

Uniquenesses:
Clay1 Clay2 Clay5  CEC1  CEC2  CEC5   OC1   OC2   OC5
0.067 0.016 0.085 0.180 0.005 0.505 0.094 0.335 0.320

Loadings:
      Factor1 Factor2 Factor3
Clay1  0.838   0.393   0.277
Clay2  0.928   0.200   0.289
Clay5  0.910   0.186   0.227
CEC1   0.144   0.797   0.404
CEC2   0.265   0.283   0.919
CEC5   0.290           0.640
OC1    0.317   0.896
OC2    0.478   0.530   0.393
OC5    0.653   0.293   0.410

               Factor1 Factor2 Factor3
SS loadings      3.322   2.116   1.955
Proportion Var   0.369   0.235   0.217
Cumulative Var   0.369   0.604   0.821

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 84 on 12 degrees of freedom.
The p-value is 7.07e-13
```

Interpretation First, the *uniqueness* of each original variable is the "noise" left over after the factors are fitted. Here CEC5 is far and away the most poorly explained, followed by OC2 and OC5.

Second, the *loadings* are just like PCA: the contribution of each original variable to the synthetic variable. Factor 1 is clearly built up mainly from all three Clay contents; Factor 2 is clearly built up from topsoil CEC and OC.

Third, the *proportional variances* are like PCA: how much of the total variance in the original set is explained by the factor. These are generally lower than the corresponding PC's.

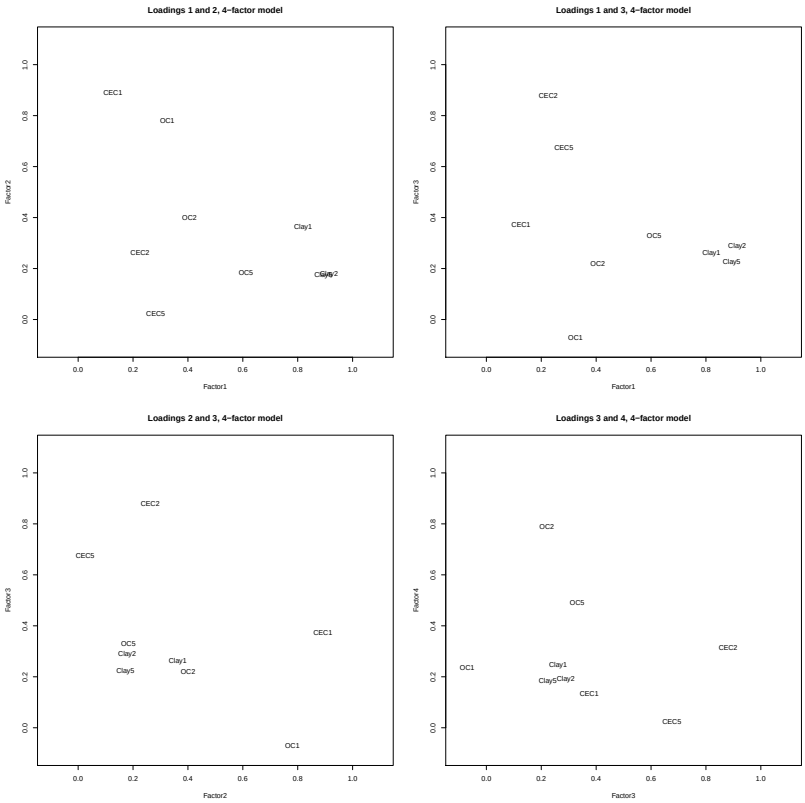
Fourth, a test of the hypothesis that the number of factors is sufficient to explain the data set. Here we see it is not, so we add another factor:

```
> (fa <- update(fa, factors = 4))

Call:
factanal(x = obs[7:15], factors = 4)

Uniquenesses:
Clay1 Clay2 Clay5  CEC1  CEC2  CEC5   OC1   OC2   OC5
0.068 0.013 0.088 0.030 0.005 0.462 0.224 0.005 0.241

Loadings:
      Factor1 Factor2 Factor3 Factor4
Clay1  0.820   0.363   0.261   0.245
Clay2  0.914   0.179   0.289   0.189
Clay5  0.894   0.172   0.224   0.185
CEC1   0.126   0.892   0.375   0.137
CEC2   0.226   0.262   0.881   0.315
CEC5   0.282           0.676
OC1    0.325   0.781           0.236
OC2    0.406   0.400   0.221   0.788
OC5    0.611   0.183   0.331   0.492

               Factor1 Factor2 Factor3 Factor4
SS loadings      3.097   1.861   1.739   1.168
Proportion Var   0.344   0.207   0.193   0.130
Cumulative Var   0.344   0.551   0.744   0.874

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 20.06 on 6 degrees of freedom.
The p-value is 0.0027
```

Now the four factors explain the data set. Notice how the uniqueness values have all decreased. The first three factors have changed somewhat.

We can visualise the meaning of the axes:

```
> par(mfrow=c(2,2))
> plot(loadings(fa), xlim=c(-.1,1.1), ylim=c(-.1,1.1), type="n",
+      main="Loadings 1 and 2, 4-factor model")
> text(loadings(fa),dimnames(loadings(fa))[[1]])
> plot(loadings(fa)[,c(1,3)], xlim=c(-.1,1.1), ylim=c(-.1,1.1), type="n",
+      main="Loadings 1 and 3, 4-factor model")
> text(loadings(fa)[,c(1,3)],dimnames(loadings(fa))[[1]])
> plot(loadings(fa)[,2:3], xlim=c(-.1,1.1), ylim=c(-.1,1.1), type="n",
+      main="Loadings 2 and 3, 4-factor model")
> text(loadings(fa)[,2:3],dimnames(loadings(fa))[[1]])
> plot(loadings(fa)[,3:4], xlim=c(-.1,1.1), ylim=c(-.1,1.1), type="n",
+      main="Loadings 3 and 4, 4-factor model")
> text(loadings(fa)[,3:4],dimnames(loadings(fa))[[1]])
> par(mfrow=c(1,1))
```



In this example, the clay contents clearly align with factor 1, high CEC and OC in the topsoil with factor 2, subsoil CEC with factor 3, and subsoil OC with factor 4.

8.3 Answers

A87 : 91.7%, 99.6%. *So the three-dimensional space is effectively two-dimensional; and even reducing to one dimension only discards about* 8% *of the total information.* *Return to Q87* •

A88 : *The first component explains much less of the overall variance in the standardized PCs, 75.6% vs. 98.4%. This is because, in the non-standardized case, one variable (clay) has much larger numbers (in range 10 – 70, mean about 30) than the others (CEC in the teens, OC below 5). Further, the units of measurement are different. If the variables had the same units of measurement and were roughly comparable (e.g. clay proportion at different depths, or analyzed by different lab. methods) the unstandardized PCA would give better insight into the relative magnitudes of variation.* *Return to Q88* •

A89 : 75.6%, 91.5%. *All three dimensions contain significant variability in the new space.* *Return to Q89* •

A90 : *Observations 78 and 81 score very low on PC1 and very high on PC 2; they have very high CEC and OC. Thus most of their CEC is accounted for by OC, not clay.*

Observation 2 scores very low on PC2 and moderately low on PC1; this has an unusually low CEC for its high clay content, because of the low OC.

Observations 3, 10, 13 and 106 score quite low on PC1 but are not exceptional for PC2.
Return to Q90 •

A91 : *The accuracy increases with the number of components used in the reconstruction. With all three, the reconstruction is exact (within numerical precision). With two, there are some fairly large errors; e.g. for* `OC1`*, as large as 4.22%. Using only one component, this increases to 6.3%. Of course, with all three there is no error.* *Return to Q91* •

A92 : *The residuals are somewhat smaller (tighter distribution about the zero-line) with the two-PC reconstruction; however in both cases they are quite related to the original values. High OC are systematically under-predicted and vice-versa. This shows that the first two PCs do not capture all the variability in OC and systematically ignore extremes. The rotations matrix shows that OC is highly related to the third component, whereas the first two components are related to clay and CEC. The systematic bias is because some high-CEC, high-clay soils have low OC and vice-versa.* *Return to Q92* •

A93 : *As with OC, the representation with one PC is poor, and has a systematic bias towards average values. But with two PCs the reconstruction is excellent. The maximum absolute-value residual is only 1.44 cmol$^+$ (kg soil)$^{-1}$ a fairly small error given the minimum observation of 11.2, although it is about 1/4 of the minimum observation (3).* *Return to Q93* •

A94 : *Of course the units of measure are different (here, standardized; in the previous analysis original units of measure). The pattern of residuals vs. original values is very similar for OC in the two cases. However for CEC the original values are reproduced very accurately and with no gain in the unstandardized case, whereas the standardized CEC is less well reproduced.* *Return to Q94* •

A95 : *There are no large clusters; indeed the data is remarkably well-distributed around* $(0, 0)$*; however the unusual observations (see next question) seem to form two small groups.* *Return to Q95* •

A96 : *Observations 81 and 78 have unusually high values of both PCs; observations 1 and 2 are nearer the centre of the range of PC1 but are unusually low for PC2. The first two have unusually high OC and CEC, with low clay; no other observations have this combination. The second two have high clay, moderate OC, but lower-than-expected CEC.* *Return to Q96* •

A97 : *All three have about the same length vectors, so are equally-well represented in t his space.* *Return to Q97* •

A98 : *CEC1 and OC1 are nearly parallel to the axis for PC1; Clay1 contributes about equally to the two axes. All three variables contribute in the same direction. So PC1 represents the general trend of all the soils in this dataset from low CEC, OC, and clay*

towards (generally correlated) high CEC, OC and clay. PC2 represents the variability in clay, and to a lesser extent CEC, not associated with this general trend. *Return to Q98* •

A99 : *CEC and OC are almost identical, implying that OC would be a good single predictor of CEC (as we saw when comparing various regressions, §7.2). Clay is about halfway from correlated to uncorrelated with these two (angle about $\pi/4$ with OC).* *Return to Q99* •

A100 : *The first PC explains almost 90% of the variance. The slope changes dramatically after the first PC and somewhat less after the third PC. Three PCs can be interpreted.* *Return to Q100* •

9 Geostatistics

These observations were made at known locations, which allows us to examine them for their *spatial structure*. First we look at the spatial distribution of the points and the data values, then we see if there is a regional *trend* and/or a *local structure*.

Note: This dataset is not ideal for spatial analysis; the sample size is not great and the sample locations are clustered. It is included here to introduce some techniques of spatial analysis.

9.1 Postplots

Task 70 : Display a map of the sample locations, coloured by agro-ecological zone, and with the symbol size proportional to the value of subsoils clay at each location (a *postplot*). •

> **Note:** Note on this code: the `asp` argument, with value `1`, to the `plot` method ensures that the horizontal and vertical axes have the same expansion, as in a map; the `grid` method draws a grid on the map.

Q101 : *Does their appear to be any regional trend in subsoil clay content? How consistent is this?* *Jump to A101* •

9.2 Trend surfaces

The regional trend can be modelled by a *trend surface*, using the grid coördinates as independent linear predictors. However, there is a problem with the naïve approach using ordinary least squares (OLS) (e.g. `lm(Clay5 ~ e + n)`, if the observations are *clustered* and, even worse, *spatially-correlated*, as seems to be the case here.

If the sample points are *clustered* in some parts of the map (as in this case), there is a danger of *mis-estimating the regression coefficients*. In particular, a large

```
> plot(e, n, cex=Clay5*3/max(Clay5), pch=20, col=as.numeric(zone), asp=1)
> grid(lty=1)
> title("Postplot of topsoil clay %, by soil type")
```

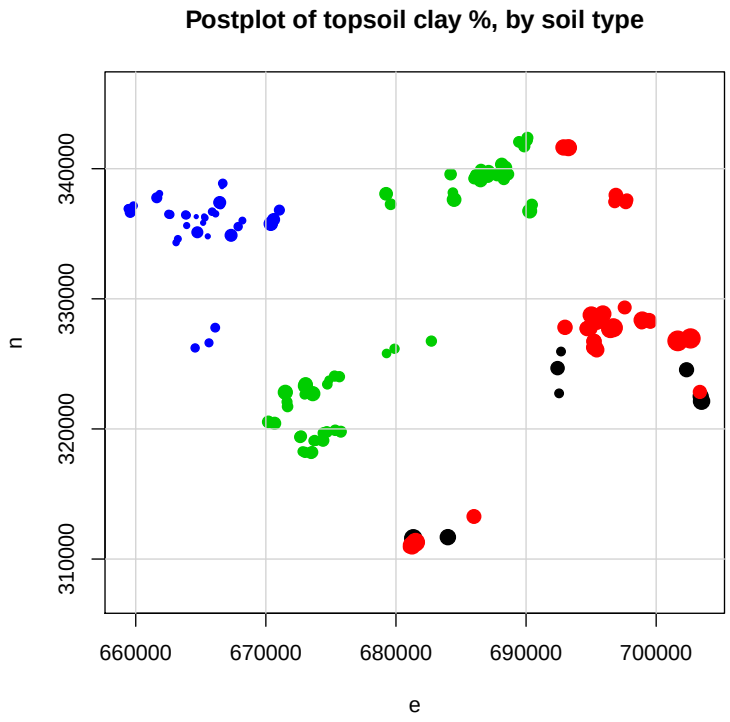


Figure 1: Postplot of clay%, 30-50 cm layer, by agro-ecological zone

number of close-by points with similar values will "pull" a trend surface towards them. Furthermore, the OLS R^2 may be over-optimistic.

The solution is to use Generalised Least Squares (GLS) to estimate the trend surface. This allows a *covariance structure between residuals* to be included directly in the least-squares solution of the regression equation. GLS is a special case of *Weighted Least Squares* (WLS).

The GLS estimate of the regression coefficients is [6]:

$$\hat{\beta}_{gls} = (X^T \cdot C^{-1} \cdot X)^{-1} \cdot X^T C^{-1} \cdot y$$

where X is the design matrix, C the *covariance matrix* of the (spatially-correlated) *residuals*, and y the observations. If there is no spatial dependence among the errors, C reduces to $I\sigma^2$ and the estimate to OLS:

$$\hat{\beta}_{ols} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

This leads us to a further difficulty: The covariance structure refers to the residuals, but we can't compute these until we fit the trend ... but we need the covariance structure to fit the trend ... which is a classic "chicken or the egg?" problem.

In practice, it is usually sufficient to (1) make a first estimate of the trend surface with OLS; (2) compute the residuals; (3) model the covariance structure of the OLS residuals as a function of their separation; (4) use this covariance structure to determine the weights to compute the GLS trend surface. The GLS residuals could again be modelled to see if their covariance structure differs from that estimated from the OLS residuals; in practice, unless the dataset is large it is not possible to see any such difference.

GLS surfaces and spatial correlation structures can both be analyzed in the `spatial` package; this follows the procedures explained by Ripley [29, 35].

Task 71 : Load the `spatial` package. Use its `surf.ls` method to compute the OLS trend surface; display its analysis of variance and coefficients. •

```
> require(spatial)
> clay5.ls<-surf.ls(1, e, n, Clay5)
> summary(clay5.ls)

Analysis of Variance Table
 Model: surf.ls(np = 1, x = e, y = n, z = Clay5)
           Sum Sq  Df  Mean Sq F value Pr(>F)
Regression  12228   2 6114.124  73.715 <2e-16
Deviation   11944 144   82.943
Total       24172 146
Multiple R-Squared: 0.506,        Adjusted R-squared: 0.499
AIC: (df = 3) 652.44
Fitted:
   Min     1Q Median     3Q    Max
  27.4   40.1   44.7   53.7   62.8
Residuals:
```

```
    Min      1Q  Median      3Q     Max
-31.601  -5.106  -0.363   3.607  20.467

> clay5.ls$beta

[1] 46.4288 14.3561 -7.0893
```

A note on trend surface coefficients computed by the `spatial` package: they do not refer to the original coördinates (`e` and `n`) but rather to *offsets* in `e` and `n` values from the centre of the trend surface area, defined by the extreme values of the coördinates. This is to make computations more stable. The first coefficient is thus the value at the centre of area:

```
> (predict(clay5.ls,
+      diff(range(e))/2 + min(e),
+      diff(range(n))/2 + min(n)))

[1] 46.429

> clay5.ls$beta[1]

[1] 46.429
```

Q102 : *How much of the variation in subsoil clay is explained by the OLS trend surface?* *Jump to A102* •

Q103 : *What is the equation of the 1st-order OLS trend surface?* *Jump to A103* •

Task 72 : Use the `correlogram` method to compute the spatial auto-correlation of subsoil clay. Examine this correlation. •

The `correlogram` method automatically computes the correlation for the residuals, once a trend surface is fit, as it was for object `clay5.ls`, above:

Q104 : *What is the autocorrelation at the shortest lag? What distance range between point-pairs is in this bin? How many point-pairs contributed to this estimate? What happens to the auto-correlation as the distance between pairs increases?* *Jump to A104* •

The observations are indeed spatially-correlated at short ranges; we now model this.

Task 73 : Fit an autocorrelation function to the correlogram and use this to fit the GLS trend surface. Display its analysis of variance and coefficients. •

This structure is not very strong and difficult to model, so we use an estimate just to show how the procedure works. An exponential model with an effective range of 1800 m seems to fit.

```
> c <- correlogram(clay5.ls, 50, plotit=F)
> str(c)
List of 3
 $ x  : num [1:48] 0 949 1898 2847 3796 ...
 $ y  : num [1:48] 0.4069 -0.1299 0.0644 0.0351 -0.1015 ...
 $ cnt: int [1:48] 429 291 299 273 345 199 116 143 127 163 ...
> plot(c, ylim=c(-.2, .6), xlim=c(0,12000), pch=20, col="blue")
> text(c$x, c$y, round(c$y, 2), pos=3)
> abline(h=0)
```

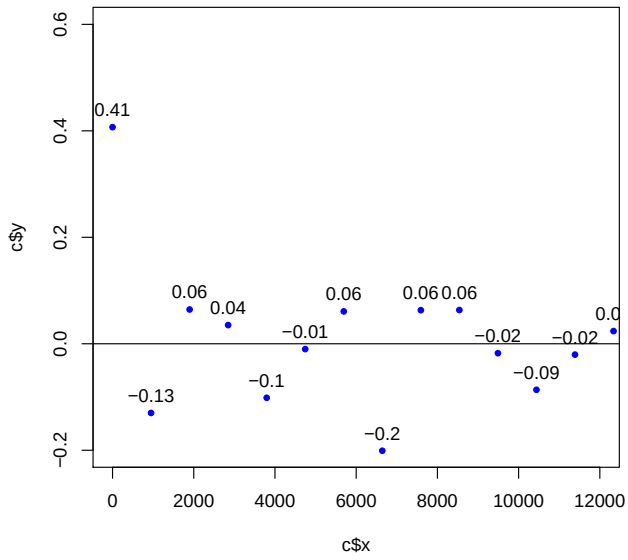
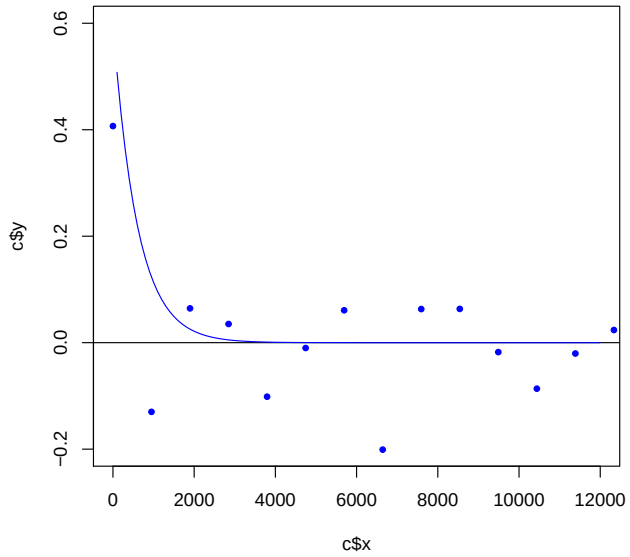


Figure 2: Auto-correlogram, clay%, 30-50cm layer

We first re-plot the correlogram, then the fit:

```
>                     # estimate function by eye: exponential
> plot(c, ylim=c(-.2, .6), xlim=c(0,12000), pch=20, col="blue")
> abline(h=0)
> d <- seq(100,12000, by=100)
> lines(d, expcov(d, d=600, alpha=.4), col="blue")
```




```
> rm(c, d)
```

We then fit a GLS trend surface, using this covariance function to model the spatial autocorrelation:

```
> # fits fairly well at close range
>                     # now fit the GLS surface
> clay5.gls<-surf.gls(1, expcov, d=600, alpha=.4, e, n, Clay5)
> summary(clay5.gls)

Analysis of Variance Table
 Model: surf.gls(np = 1, covmod = expcov, x = e, y = n, z = Clay5, d = 600,     alpha
           Sum Sq  Df  Mean Sq F value Pr(>F)
Regression  12150   2 6075.201  72.772 <2e-16
Deviation   12022 144   83.483
Total       24172 146
Multiple R-Squared: 0.503,        Adjusted R-squared: 0.496
AIC: (df = 3) 653.39
Fitted:
   Min     1Q Median     3Q    Max
  27.8   41.0   45.0   54.4   63.5
Residuals:
   Min     1Q Median     3Q    Max
-32.40  -5.61  -1.03   3.14  19.58

> clay5.gls$beta
```

```
[1] 46.8158 14.8960 -6.3662
```

Q105 : *How do the R^2 and coefficients compare to the OLS surface?* *Jump to A105* •

Task 74 : Plot the OLS and GLS trend surfaces, with the sample points superimposed. •

We use the `eqscplot` method of the `MASS` library, as well as the `contourplot` method of the `lattice` library; both these must be loaded; `MASS` was loaded above, so here we just load `lattice`:

```
> require(lattice)
```

Task 75 : Make a postplot of the residuals, with the symbol coloured according to whether it is positive or negative. •

Q106 : *Does there appear to be any spatial pattern to the residuals?* *Jump to A106* •

9.3 Higher-order trend surfaces

Evidently the first-order trend surface (a plane) captured a regional trend, but the fit was not very good ($R^2 = 0.496$). This suggests that a *higher-order surface* may be more satisfactory, both mathematically and in explaining the trend. That is, the trend may not be a plane, but rather a second-order surface such as a dome or saddle. In this case the residuals did not show an obvious pattern to suggest this, but still we will try.

Task 76 : Compute and plot a 2^{nd}-order trend surface and summarize its goodness-of-fit. •

Q107 : *How well does this trend surface fit the observations? Is this an improvement over the 1^{st}-order surface?* *Jump to A107*
•

Q108 : *Describe the form of the 2^{nd}-order surface. Does its main 1^{st}-order axis match the 1^{st}-order surface?* *Jump to A108* •

9.4 Local spatial dependence and Ordinary Kriging

In the previous two sections we've considered a regional ("global") trend in subsoil clay. However it is evident that there is *local spatial autocorrelation*, that is,

```
> xmin <- min(e); xmax <- max(e); ymin <- min(n); ymax <- max(n); res <- 40
> clay5.ts <- trmat(clay5.ls, xmin, xmax, ymin, ymax, res)
> clay5.gts <- trmat(clay5.gls, xmin, xmax, ymin, ymax, res)
> eqscplot(clay5.gts, type="n",
+      main="OLS and GLS trend surfaces, subsoil clay %", xlab="E", ylab="N")
> contour(clay5.gts, level=seq(20, 80, 4), add=T)
> contour(clay5.ts, level=seq(20, 80, 4), add=T, lty=2, col="blue")
> grid(lty=1)
> points(e, n, cex=Clay5*2.5/max(Clay5), pch=23, bg=3)
> rm(clay5.ts, clay5.gts, xmin, xmax, ymin, ymax, res)
```

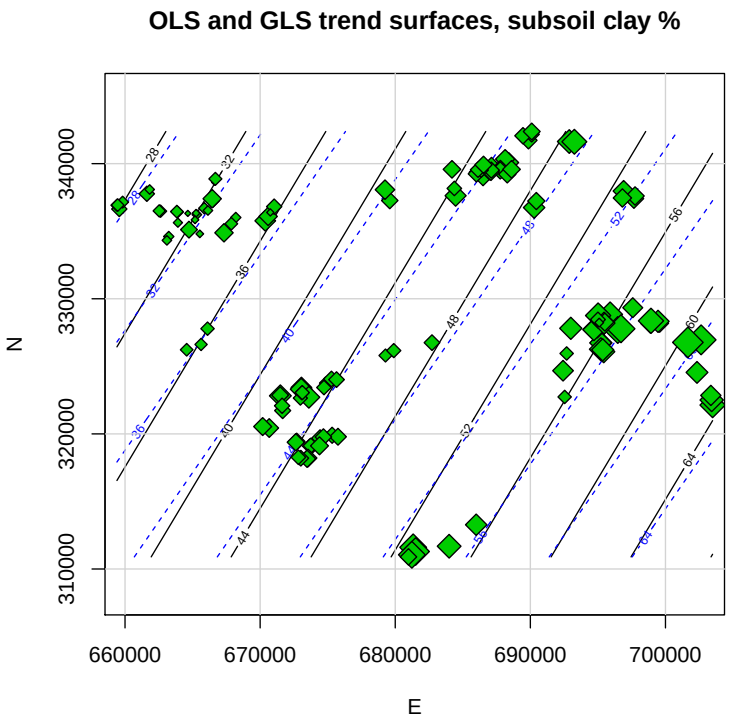


Figure 3: First-order OLS and GLS trend surfaces for clay%, 30-50cm layer

```
> xmin <- min(e); xmax <- max(e); ymin <- min(n); ymax <- max(n); res <- 40
> clay5.gls <- surf.gls(1, expcov, d=600, alpha=.4, e, n, Clay5)
> clay5.gls.resid <- resid(clay5.gls)
> clay5.gts <- trmat(clay5.gls, xmin, xmax, ymin, ymax, res)
> eqscplot(clay5.gts, type="n",
+      main="Residuals from GLS 1st-order trend surface, subsoil clay %",
+      sub="Red: negative; Green: positive",
+      xlab="E", ylab="N")
> contour(clay5.gts, level=seq(20, 80, 4), add=T)
> grid(lty=1)
> points(e, n, cex=abs(clay5.gls.resid)*2.5/max(abs(clay5.gls.resid)),
+        pch=23, bg=ifelse(clay5.gls.resid < 0, 3, 2))
> rm(clay5.gls, clay5.gts, clay5.gls.resid, xmin, xmax, ymin, ymax, res)
```

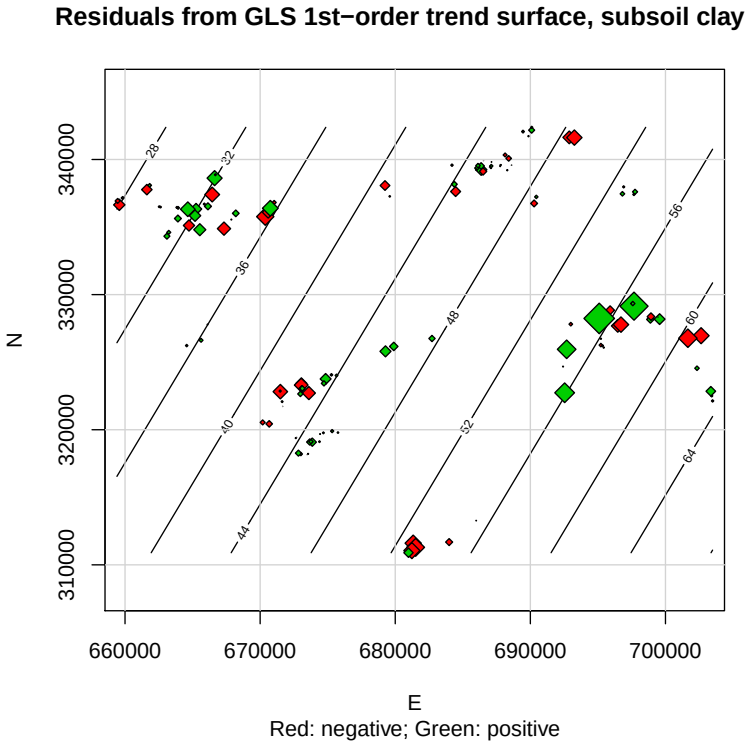


Figure 4: Residuals from first-order GLS trend surface for clay%, 30-50cm layer

```
> xmin <- min(e); xmax <- max(e); ymin <- min(n); ymax <- max(n); res <- 40
> clay5.gls2 <- surf.gls(2, expcov, d=600, alpha=.4, e, n, Clay5)
> summary(clay5.gls2)

Analysis of Variance Table
 Model: surf.gls(np = 2, covmod = expcov, x = e, y = n, z = Clay5, d = 600,     alpha = 0.4)
           Sum Sq  Df  Mean Sq F value Pr(>F)
Regression  12832   5 2566.460  31.912 <2e-16
Deviation   11340 141   80.423
Total       24172 146
Multiple R-Squared: 0.531,        Adjusted R-squared: 0.514
AIC: (df = 6) 650.81
Fitted:
   Min     1Q Median     3Q    Max
  30.3   39.0   45.4   52.8   64.5
Residuals:
   Min     1Q Median     3Q    Max
-29.86  -5.35  -1.11   3.85  20.39

> clay5.gts2 <- trmat(clay5.gls2, xmin, xmax, ymin, ymax, res)
> eqscplot(clay5.gts2, type="n",
+      main="GLS 2nd-order trend surface, subsoil clay %", xlab="E", ylab="N")
> contour(clay5.gts2, level=seq(20, 80, 4), add=T)
> grid(lty=1)
> points(e, n, cex=Clay5*2.5/max(Clay5), pch=23, bg=3)
> clay5.gls2$beta

[1] 42.2565 13.6599  6.8285 -8.7301  5.0466  8.9743

> rm(clay5.gls2, clay5.gts2, xmin, xmax, ymin, ymax, res)
```

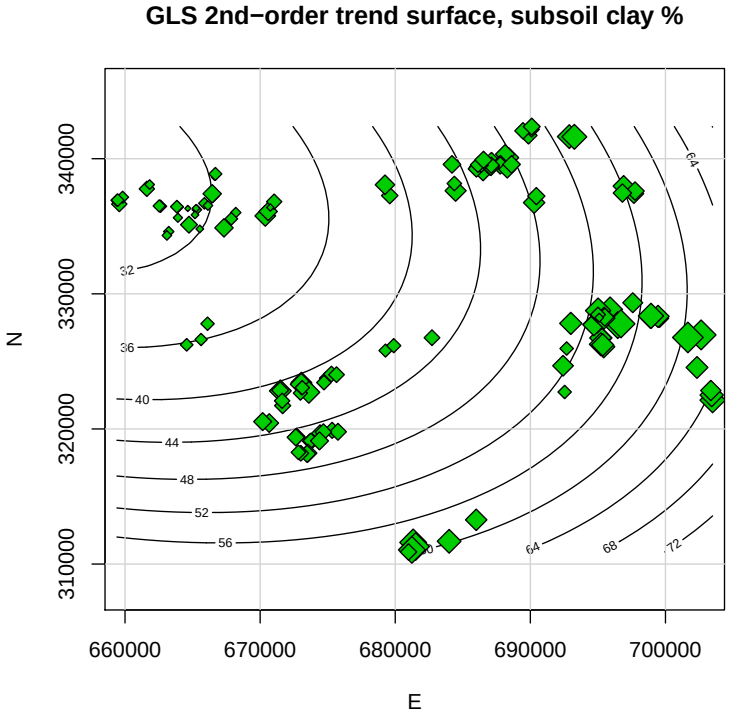


Figure 5: Second-order GLS trend surface for clay%, 30-50cm layer

nearby points tend to be similar. The correlogram of Figure 2 shows this spatial autocorrelation. This is best analyzed with *variograms*; the results can be used for mapping by "optimal" interpolation: *Ordinary Kriging*. R has several packages for this; we will use `gstat` [25–27] which is part of the spatial data initiative, and uses the `sp` "spatial classes" package.

Task 77 : Load the `gstat` and `sp` packages with the `library` method and confirm they are in the search path, with the `search` method. •

```
> require(sp)
> require(gstat)
> search()

 [1] ".GlobalEnv"        "package:gstat"     "package:sp"
 [4] "package:lattice"   "package:spatial"   "obs"
 [7] "package:car"       "package:MASS"      "ESSR"
[10] "package:stats"     "package:graphics"  "package:grDevices"
[13] "package:utils"     "package:datasets"  "package:methods"
[16] "Autoloads"         "package:base"
```

9.4.1 Spatially-explicit objects

Task 78 : Review the structure of the `obs` object. •

```
> str(obs)

'data.frame':        147 obs. of  15 variables:
 $ e    : int  702638 701659 703488 703421 703358 702334 681328 681508 681230 683989
 $ n    : int  326959 326772 322133 322508 322846 324551 311602 311295 311053 311685
 $ elev : int  657 628 840 707 670 780 720 657 600 720 ...
 $ zone : Factor w/ 4 levels "1","2","3","4": 2 2 1 1 2 1 1 2 2 1 ...
 $ wrb1 : Factor w/ 3 levels "a","c","f": 3 3 3 3 3 3 3 3 3 3 ...
 $ LC   : Factor w/ 8 levels "BF","CF","FF",..: 3 3 4 4 4 4 3 3 4 4 ...
 $ Clay1: int  72 71 61 55 47 49 63 59 46 62 ...
 $ Clay2: int  74 75 59 62 56 53 66 66 56 63 ...
 $ Clay5: int  78 80 66 61 53 57 70 72 70 62 ...
 $ CEC1 : num  13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6 7.9 14.9 ...
 $ CEC2 : num  10.1 8.2 10.2 8.4 9.2 11.6 7.4 7.1 5.7 6.8 ...
 $ CEC5 : num  7.1 7.4 6.6 8 8.5 6.2 5.4 7 4.5 6 ...
 $ OC1  : num  5.5 3.2 6.98 3.19 4.4 5.31 4.55 4.5 2.3 7.34 ...
 $ OC2  : num  3.1 1.7 2.4 1.5 1.2 3.2 2.15 1.42 1.36 2.54 ...
 $ OC5  : num  1.5 1 1.3 1.26 0.8 ...
```

Q109 : *What is the data type of this object? Which of the fields refer to* spatial *information? What is their data type?* *Jump to A109* •

The data types for the `e` and `n` fields in the data frame are `int`, i.e. integers. These are indeed numbers, but of a special kind: they are *coördinates* in geographic space.

It is possible to do some visualization and analysis in R with the data frame, but it is more elegant, and gives many more possibilities, if geographic data is explicitly recognized as such. This was the motivation behind the *R Spatial Project*, which resulted in the `sp` package [24] which provides *classes* (data types and methods for these) for spatial data.

The `sp` package adds a number of *spatial data types*, i.e. new *object classes*; these are then recognized by methods in other packages that are built on top of `sp`, most notably (for our purposes) the `gstat` package.

To take advantage of the power of an explicit spatial representation, we must convert the data frame to the most appropriate `sp` class.

Task 79 : Create a new object of class `SpatialPointsDataFrame` named `obs.sp`, from the `obs` dataframe. •

We do this by adding the computed coordinates to the data frame with the `coordinates` method; this automatically converts to the spatial data type defined by the `sp` package:

```
> class(obs)

[1] "data.frame"

> obs.sp <- obs
> coordinates(obs.sp) <- ~ e + n
> class(obs.sp)

[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"
```

This is a syntax we haven't seen before; the `coordinates` method can appear either on the right or the left of the assignment operator, and the formula operators `~` and `+`

Q110 : *What is the data type of the `obs.sp` object?* *Jump to A110* •

Task 80 : View the structure and data summary of the spatial object. •

As usual, the structure is displayed by the `str` method; we then summarize the object with the generic `summary` method and view the first few records in the dataframe with the `head` method:

```
> str(obs.sp)

Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
  ..@ data       :'data.frame':        147 obs. of  13 variables:
  .. ..$ elev : int [1:147] 657 628 840 707 670 780 720 657 600 720 ...
  .. ..$ zone : Factor w/ 4 levels "1","2","3","4": 2 2 1 1 2 1 1 2 2 1 ...
  .. ..$ wrb1 : Factor w/ 3 levels "a","c","f": 3 3 3 3 3 3 3 3 3 3 ...
  .. ..$ LC   : Factor w/ 8 levels "BF","CF","FF",..: 3 3 4 4 4 4 3 3 4 4 ...
```

```
  .. ..$ Clay1: int [1:147] 72 71 61 55 47 49 63 59 46 62 ...
  .. ..$ Clay2: int [1:147] 74 75 59 62 56 53 66 66 56 63 ...
  .. ..$ Clay5: int [1:147] 78 80 66 61 53 57 70 72 70 62 ...
  .. ..$ CEC1 : num [1:147] 13.6 12.6 21.7 11.6 14.9 18.2 14.9 14.6 7.9 14.9 ...
  .. ..$ CEC2 : num [1:147] 10.1 8.2 10.2 8.4 9.2 11.6 7.4 7.1 5.7 6.8 ...
  .. ..$ CEC5 : num [1:147] 7.1 7.4 6.6 8 8.5 6.2 5.4 7 4.5 6 ...
  .. ..$ OC1  : num [1:147] 5.5 3.2 6.98 3.19 4.4 5.31 4.55 4.5 2.3 7.34 ...
  .. ..$ OC2  : num [1:147] 3.1 1.7 2.4 1.5 1.2 3.2 2.15 1.42 1.36 2.54 ...
  .. ..$ OC5  : num [1:147] 1.5 1 1.3 1.26 0.8 ...
  ..@ coords.nrs : int [1:2] 1 2
  ..@ coords     : num [1:147, 1:2] 702638 701659 703488 703421 703358 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:147] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "e" "n"
  ..@ bbox       : num [1:2, 1:2] 659401 310897 703488 342379
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "e" "n"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
  .. .. ..@ projargs: chr NA
```

```
> head(obs.sp@data)

  elev zone wrb1 LC Clay1 Clay2 Clay5 CEC1 CEC2 CEC5  OC1 OC2  OC5
1  657    2    f FF    72    74    78 13.6 10.1  7.1 5.50 3.1 1.50
2  628    2    f FF    71    75    80 12.6  8.2  7.4 3.20 1.7 1.00
3  840    1    f FV    61    59    66 21.7 10.2  6.6 6.98 2.4 1.30
4  707    1    f FV    55    62    61 11.6  8.4  8.0 3.19 1.5 1.26
5  670    2    f FV    47    56    53 14.9  9.2  8.5 4.40 1.2 0.80
6  780    1    f FV    49    53    57 18.2 11.6  6.2 5.31 3.2 1.08
```

```
> summary(obs.sp@data)

      elev      zone   wrb1          LC         Clay1
 Min.   :  82   1: 8   a: 40   FV     :69   Min.   :10.0
 1st Qu.: 322   2:40   c:  3   BF     :19   1st Qu.:21.0
 Median : 432   3:63   f:104   FF     :17   Median :30.0
 Mean   : 418   4:36           CF     :15   Mean   :31.3
 3rd Qu.: 560                  OCA    :14   3rd Qu.:39.0
 Max.   :1000                  MCA    :11   Max.   :72.0
                               (Other): 2
     Clay2          Clay5           CEC1           CEC2
 Min.   : 8.0   Min.   :16.0   Min.   : 3.0   Min.   : 1.60
 1st Qu.:27.0   1st Qu.:36.5   1st Qu.: 7.5   1st Qu.: 5.00
 Median :36.0   Median :44.0   Median :10.1   Median : 7.00
 Mean   :36.7   Mean   :44.7   Mean   :11.2   Mean   : 7.41
 3rd Qu.:47.0   3rd Qu.:54.0   3rd Qu.:13.1   3rd Qu.: 9.40
 Max.   :75.0   Max.   :80.0   Max.   :29.0   Max.   :22.00

      CEC5            OC1             OC2            OC5
 Min.   : 1.00   Min.   : 1.04   Min.   :0.30   Min.   :0.20
 1st Qu.: 5.00   1st Qu.: 1.98   1st Qu.:0.85   1st Qu.:0.60
 Median : 6.50   Median : 2.70   Median :1.30   Median :0.84
 Mean   : 6.84   Mean   : 2.99   Mean   :1.39   Mean   :0.81
 3rd Qu.: 8.90   3rd Qu.: 3.70   3rd Qu.:1.70   3rd Qu.:1.00
 Max.   :14.00   Max.   :10.90   Max.   :3.70   Max.   :1.70
```

The spatial object now has several *slots*, marked with the `@` symbol in the structure listing.

9.4.2 Analysis of local spatial structure

With this preparation, we can now use the `gstat` package to analyze local spatial structure.

Task 81 : Compute and plot the *empirical variogram* of subsoil clay. •

```
> v <- variogram(Clay5 ~ 1, obs.sp)
> print(plot(v, pl=T, pch=20, col="blue", cex=1.5))
```

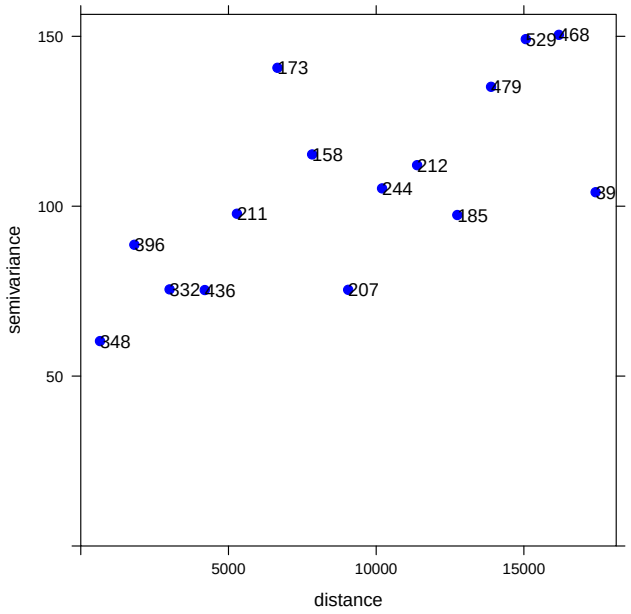


Figure 6: Experimental variogram for clay%, 30-50cm layer

The experimental variogram is shown in Figure 6. Note that the `pl=T` argument to the `plot` method shows the number of point-pairs for each estimate.

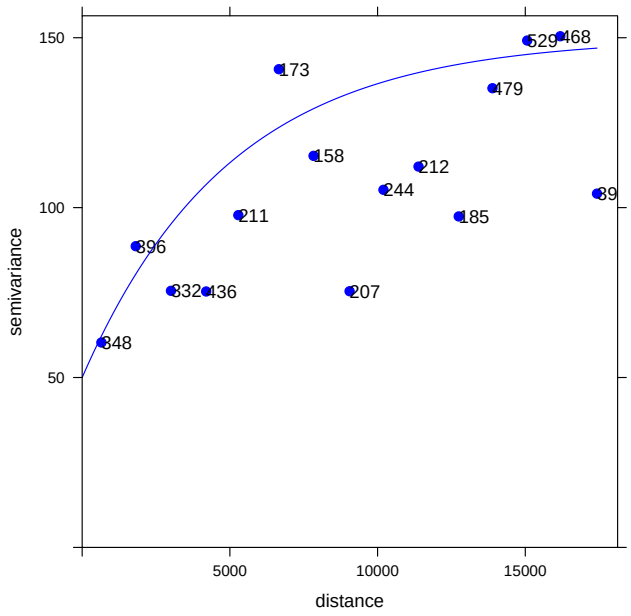
Q111 : *Does there seem to be local spatial dependence? What is the evidence for this?* *Jump to A111* •

Q112 : *What is an appropriate variogram shape and approximate range, sill, and nugget?* *Jump to A112* •

Task 82 : Estimate (by eye) a variogram model to fit the experimental variogram. •

Note that the range *parameter* of an exponential model is 1/3 of the effective range.

```
> m <- vgm(100, "Exp", 15000/3, 50)
> print(plot(v, pl=T, pch=20, col="blue", cex=1.5, model=m))
```



Task 83 : Fit a variogram model to the experimental variogram, using the `gstat` `fit.variogram` method. •

The fitted model is shown superimposed on the experimental variogram in Figure 7.

Q113 : *What are the parameters of your estimate and the best fit as determined by* `gstat`*?* *Jump to A113* •

9.4.3 Interpolation by Ordinary Kriging

Once we have a model of the local spatial structure, we can use this to map the study area by *kriging*, which, if the model is correct, is an *optimal interpolator*.

> **Note:** This dataset is not really suitable for interpolation, since there are large areas far from any sample points. The samples do not have to be evenly-distributed, but there should be some points within variogram range of all areas to be interpolated; with the fitted exponential model, this range is about $3 * 2626 \approx 8000$ meters. We will see the effect of this irregular point distribution in the map of kriging prediction variance.

Task 84 : Use this variogram model to interpolate across the study area by Ordinary Kriging; also produce a map of the kriging prediction variance. •

```
> (m.f <- fit.variogram(v, m))
  model  psill  range
1   Nug 47.188    0.0
2   Exp 62.349 2626.5
> str(m.f)
Classes 'variogramModel' and 'data.frame':        2 obs. of  9 variables:
 $ model: Factor w/ 20 levels "Nug","Exp","Sph",..: 1 2
 $ psill: num  47.2 62.3
 $ range: num  0 2626
 $ kappa: num  0 0.5
 $ ang1 : num  0 0
 $ ang2 : num  0 0
 $ ang3 : num  0 0
 $ anis1: num  1 1
 $ anis2: num  1 1
 - attr(*, "singular")= logi FALSE
 - attr(*, "SSErr")= num 0.0488
 - attr(*, "call")= language fit.variogram(object = v, model = m)
> attr(m.f, "SSErr")
[1] 0.048754
> print(plot(v, pl=T, pch=20, col="blue", cex=1.5, model=m.f))
```

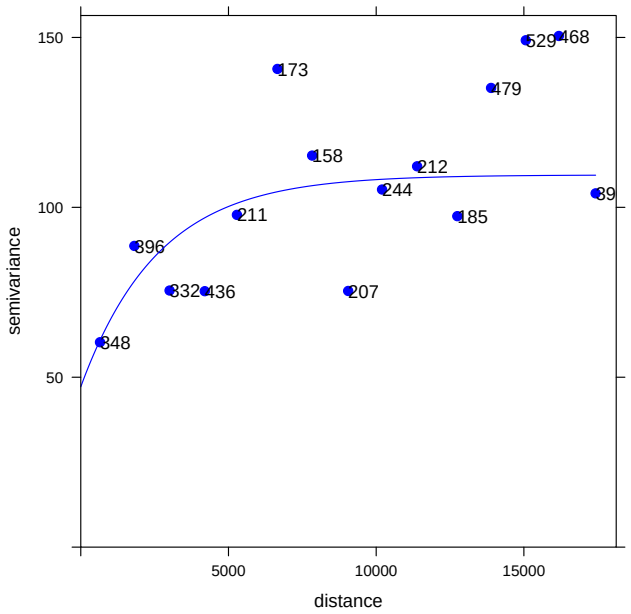


Figure 7: Experimental variogram, with fitted exponential model, for clay%, 30-50cm layer

First we have to create a grid onto which to interpolate. We do this from the bounding box of the study area.

The overall dimensions are (in km, and km^2):

```
> diff(range(e))/1000

[1] 44.087

> diff(range(n))/1000

[1] 31.482

> diff(range(e)) * diff(range(n)) / 10^6

[1] 1387.9
```

So a 1 x 1 km grid would require about 1388 cells; we'll use double that resolution, i.e. 500 x 500 m.

We first use the `expand.grid` method to make the grid, then `coordinates` to make it a spatial object, and finally `gridded` to specify that this is a regular spatial grid, not just a collection of points:

```
> res <- 500
> g500 <- expand.grid(e = seq(min(e), max(e), by=res), n = seq(min(n), max(n), by=res
> coordinates(g500) <- ~ e + n
> gridded(g500) <- T
> str(g500)

Formal class 'SpatialPixels' [package "sp"] with 5 slots
  ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 659401 310897
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cellsize         : Named num [1:2] 500 500
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cells.dim        : Named int [1:2] 89 63
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  ..@ grid.index : int [1:5607] 5519 5520 5521 5522 5523 5524 5525 5526 5527 5528 ...
  ..@ coords     : num [1:5607, 1:2] 659401 659901 660401 660901 661401 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5607] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "e" "n"
  ..@ bbox       : num [1:2, 1:2] 659151 310647 703651 342147
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "e" "n"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
  .. .. ..@ projargs: chr NA

> rm(res)
```

Now we krige onto this grid, and display the prediction and variance maps:

```
> k.o <- krige(Clay5 ~ 1, obs.sp, g500, m.f)

[using ordinary kriging]
```

```
> str(k.o)

Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots
  ..@ data       :'data.frame':        5607 obs. of  2 variables:
  .. ..$ var1.pred: num [1:5607] 46.3 46.3 46.3 46.3 46.3 ...
  .. ..$ var1.var : num [1:5607] 113 113 113 113 113 ...
  ..@ coords.nrs : int [1:2] 1 2
  ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 659401 310897
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cellsize         : Named num [1:2] 500 500
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cells.dim        : Named int [1:2] 89 63
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  ..@ grid.index : int [1:5607] 5519 5520 5521 5522 5523 5524 5525 5526 5527 5528 ...
  ..@ coords     : num [1:5607, 1:2] 659401 659901 660401 660901 661401 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5607] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "e" "n"
  ..@ bbox       : num [1:2, 1:2] 659401 310897 703401 341897
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "e" "n"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
  .. .. ..@ projargs: chr NA
```

The maps are shown in Figure 8: predictions left and kriging prediction variances right.

Q114 : *Does this map seem realistic?* *Jump to A114* •

Q115 : *Explain the pattern of prediction variances.* *Jump to A115* •

Block OK Although we predicted on a 500x500 m grid, the reported prediction variances refer to a plot of the same size as the original sample, also called its *support*. Here that is not specified exactly, but we know it is a small plot, about 0.5 ha (≈ 70x70 m); this is smaller than the grid size. Note that samples were bulked from the whole field; although each sample is just one auger core, if they are taken from different places in the field and mixed, it is as if the entire soil were mixed and then subsampled. Thus the reported kriging prediction variance is for an 0.5 plot on 500 m centres.

If we are satisfied with average values over a larger area, we should use *block kriging* at that size; this determines an average, rather than point, value, and reduces the kriging prediction variance, because all variation smaller than the block is ignored. This is easy to do with the `krige` method, simply by specifying the block size.

Task 85 : Predict by ordinary kriging in 500 m by 500 m blocks, and compute

```
> plot.1 <- spplot(k.o, zcol="var1.pred",
+    main="OK prediction of Clay %, 30-50 cm", col.regions=bpy.colors(128),
+                  pretty=T)
> plot.2 <- spplot(k.o, zcol="var1.var",
+    main="OK prediction variance of Clay %, 30-50 cm", col.regions=cm.colors(128),
+                  pretty=T)
> print(plot.1, split=c(1,1,2,1), more=T)
> print(plot.2, split=c(2,1,2,1), more=F)
```

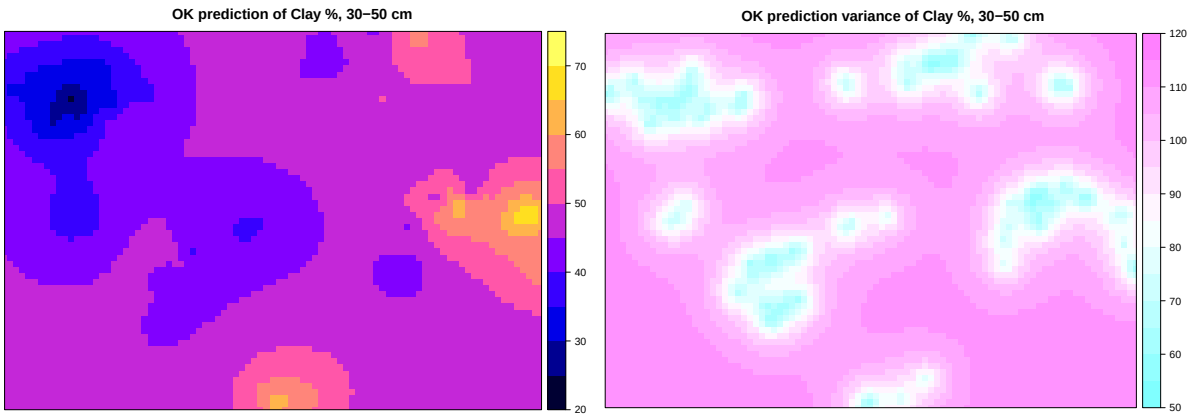


Figure 8: Predictions and standard deviation of the prediction, clay%, 30-50cm layer

the difference in predicted values and variances. •

```
> k.o.500 <- krige(Clay5 ~ 1, obs.sp, g500, m.f, block=c(500, 500))

[using ordinary kriging]

> str(k.o.500)

Formal class 'SpatialPixelsDataFrame' [package "sp"] with 7 slots
  ..@ data       :'data.frame':        5607 obs. of  2 variables:
  .. ..$ var1.pred: num [1:5607] 46.3 46.3 46.3 46.3 46.3 ...
  .. ..$ var1.var : num [1:5607] 60.2 60.2 60.2 60.2 60.2 ...
  ..@ coords.nrs : int [1:2] 1 2
  ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
  .. .. ..@ cellcentre.offset: Named num [1:2] 659401 310897
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cellsize         : Named num [1:2] 500 500
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  .. .. ..@ cells.dim        : Named int [1:2] 89 63
  .. .. .. ..- attr(*, "names")= chr [1:2] "e" "n"
  ..@ grid.index : int [1:5607] 5519 5520 5521 5522 5523 5524 5525 5526 5527 5528 ...
  ..@ coords     : num [1:5607, 1:2] 659401 659901 660401 660901 661401 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:5607] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "e" "n"
  ..@ bbox       : num [1:2, 1:2] 659401 310897 703401 341897
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "e" "n"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
  .. .. ..@ projargs: chr NA

> summary(k.o$var1.pred - k.o.500$var1.pred)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.61300 -0.00124  0.00007  0.00037  0.00135  0.76000

> summary(k.o.500$var1.var / k.o$var1.var)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.103   0.446   0.500   0.468   0.519   0.532
```

The predictions are almost the same but the prediction variances are much smaller, from 1/8 to 1/2 of those for punctual ordinary Kriging.

9.5 Answers

A101 : *See Figure 1.*

Subsoil clay appears to increase from the NW to the SE. There are local anomalies to this trend, especially in zone 2 (NW corner). *Return to Q101* •

A102 : *Almost half of the variation is explained by the trend surface. Residuals are symmetric, with most within 5% clay, but a few are quite large, both positive and*

negative. This means that the surface does not account for some local variability. *Return to Q102* •

A103 : $z = 42.4288 + 14.3561 \cdot x - 7.0893 \cdot y$, *where* x *and* y *are offsets from the centre of the area.* *Return to Q103* •

A104 : *See Figure 2.*

There were 429 point-pairs with a separation from 0 *to* $949/2 \approx 475$ *m. Their autocorrelation was* 0.41*. At the next bin, the autocorrelation has decreased to near zero, and stays there, fluctuating irregularly around zero as distance increases.* *Return to Q104* •

A105 : *See Figure 3.*

The R^2 *is slightly lower (more realistic, accounting for spatial correlation); the coefficients changed slightly (OLS:* $46.43, 14.36, -7.09$*; GLS:* $46.8214.90 - 6.37$*): a bit more N-S trend and a bit less E-W; i.e. the direction of the trend surface rotates slightly towards the S, from N107.2°E to N107.7°E.*

In this case the GLS and OLS trend surfaces are almost the same, which can be appreciated in the plot. *Return to Q105*
•

A106 : *See Figure 4.*

There is no clear pattern; some large positive and negative residuals are close to each other, so the fit can't be improved. There are some small clusters of large residuals with the same sign, but with no obvious trend. *Return to Q106* •

A107 : *The* 2^{nd}*-order* $R^2 = 0.514$*, which is only a bit higher than the* 1^{st}*-order* $R^2 = 0.496$*, i.e.1.8% more variance was explained.* *Return to Q107* •

A108 : *See Figure 5.*

The 1^{st}*-order plane, trending to the ESE, is preserved; the* 2^{nd}*-order structure is domed in the middle of this trend and falls off to the NNE and SSW. Thus the* 2^{nd}*-order trend improves but does not fundamentally change the* 1^{st}*-order trend. Note that the linear coefficients of the* 2^{nd}*-order trend* $(13.6599, -8.7301)$ *are similar to the* 1^{st}*-order trend* $(14.8960, -6.3662)$*.* *Return to Q108* •

A109 : *The object is a data frame. Fields* `e` *and* `n` *give the UTM coordinates; field* `elev` *gives the third (elevation) coordinate. These three are all integers (i.e. whole numbers). The other fields are measured attributes.* *Return to Q109* •

A110 : *The data type is now* `SpatialPointsDataFrame`*; this is defined in the* ***sp*** *package.* *Return to Q110* •

A111 : *Spatial information includes (1)* a bounding box, *i.e. limits of the the coördinates; (2) the geographic* projection, *in this case marked as* `NA` *("not applicable") because we haven't informed* `sp` *about it.* *Return to Q111* •

A112 : *Non-spatial information is the data frame less the coördinates, i.e. all the feature (attribute) space information: the two categorical variables and the seven metal contents.* *Return to Q112* •

A113 : *Yes; the evidence is that semivariances at close separation distances between point pairs are generally lower than those at longer distances.* *Return to Q113* •

A114 : *This variogram is erratic and difficult to model, mainly because of the low number of point pairs at some separations. It does not seem to reach a clear sill, so an exponential model (which approaches a sill asymptotically) may be most appropriate. The nugget is about* $50\%^2$*; the sill somewhere near* $150\%^2$*, and the effective range 15 km.* *Return to Q114* •

A115 : *The estimated parameters of the exponential model were: partial sill 100, nugget 50, range 5 000 (n.b. this implies an effective range of 15 000). The fitting algorithm emphasis close-separation points and large numbers of point pairs, and thus lowered the partial sill and shortened the range considerably: partial sill 62.3, nugget 47.2, range 2 627 (effective range 7 881).* *Return to Q115* •

A116 : *The map respects the general clusters of higher or lower values (also seen on the* 2^{nd}*-order trend surface) but seems to have small patches that are artefacts of the sampling. Away from the sample cluster, the spatial mean is predicted, with no detail.* *Return to Q116* •

A117 : *Prediction variances are low near the point clusters; in areas further than the effective range (about 8 km) the variance is maximum.* *Return to Q117* •

10 Going further

The techniques introduced in this note do not even begin to exhaust the possibilities offered by the R environment. There are several good R-specific texts which you can consult:

- Dalgaard [7] is a simple introduction to elementary statistics using R for the examples. If you found these notes too advanced, this is a good book to get you started.
- Fox [12] is a thorough treatments of applied regression and linear models, with examples from social and political sciences. It is rigorous but provides many aids for understanding. This is accompanied by a text with all the techniques illustrated by R code [13].
- Venables and Ripley [35] is the well-known *Modern applied statistics with S*, which gave rise to the `MASS` package. This has a wealth of advanced

techniques, but also some well worked-out examples of statistical modelling. Some topics covered are linear modelling, multivariate analysis, spatial statistics, and temporal statistics. I highly recommend this book for serious S users.

And remember: Each R package has its own documentation and demonstrations. There are a large number of specialised packages; one may be exactly what you need. Browse through CRAN and the R help and package archives.

Task Views

A useful R resource is the set of **Task Views**, on-line at `http://cran.r-project.org/src/contrib/Views/`. These are a summary by a task maintainer of the facilities in R to accomplish certain tasks, with links to all relevant packages. In particular there is a task view for "Multivariate Statistics" at `http://cran.r-project.org/src/contrib/Views/Multivariate.html`.

The best resource is a good statistics textbook at your level; anything explained in these is either already available in R or can be directly programmed.

Above all, *experiment* and *keep thinking*!

References

[1] E. C. Barrett and L. F. Curtis. *Introduction to environmental remote sensing.* Stanley Thornes Publishers, Cheltenham, Glos., UK, 4th edition, 1999. 99

[2] D Birkes and Y Dodge. *Alternative methods of regression.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York, 1993. 42

[3] KA Brownlee. *Statistical theory and methodology in science and engineering.* John Wiley & Sons, New York, 2nd edition, 1965. 2

[4] MG Bulmer. *Principles of statistics.* Dover Publications, New York, 1979. 2

[5] RD Cook and S Weisberg. *Residuals and influence in regression.* Chapman and Hall, New York, 1982. 147

[6] N Cressie. *Statistics for spatial data.* John Wiley & Sons, revised edition, 1993. 121

[7] P Dalgaard. *Introductory Statistics with R.* Springer Verlag, 2002. 1, 2, 140

[8] JC Davis. *Statistics and data analysis in geology.* Wiley, New York, 1986. 45, 46

[9] JC Davis. *Statistics and data analysis in geology.* John Wiley & Sons, New York, 3rd edition, 2002. 2, 45

[10] P Driessen, J Deckers, O Spaargaren, and F Nachtergaele, editors. *Lecture notes on the major soils of the world.* World Soil Resources Report 94. FAO, Rome, 2001. 3

[11] FAO. *World Reference Base for Soil Resources.* World Soil Resources Report 84. FAO; ISRIC, Rome; Wageningen, 1998. 3

[12] J Fox. *Applied regression, linear models, and related methods*. Sage, Newbury Park, 1997. 1, 83, 140

[13] J Fox. *An R and S-PLUS Companion to Applied Regression*. Sage, Newbury Park, 2002. 83, 140

[14] John Fox and Harvey Sanford Weisberg. *An R companion to applied regression*. SAGE Publications, Inc, second edition edition, Nov 2010. ISBN 978-1-4129-7514-8. 1

[15] JC Gower and DJ Hand. *Biplots*. Monographs on statistics and applied probability ; 54. Chapman & Hall, London ; New York, 1996. 109

[16] R Ihaka and R Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996. 1

[17] P Legendre and L Legendre. *Numerical ecology*. Elsevier Science, Oxford, 2nd english edition, 1998. 2, 109

[18] F Leisch. Sweave, part I: Mixing R and LaTeX. *R News*, 2(3):28–31, December 2002. URL http://CRAN.R-project.org/doc/Rnews/. 3

[19] F Leisch. *Sweave User's Manual*. TU Wein, Vienna (A), 2.1 edition, 2006. URL http://www.ci.tuwien.ac.at/~leisch/Sweave. 3

[20] Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman. *Remote sensing and image interpretation*. John Wiley & Sons, Hoboken, NJ, 6th edition, 2007. 99

[21] DM Mark and M Church. On the misuse of regression in earth science. *Mathematical Geology*, 9(1):63–77, 1977. 16

[22] MB McBride. *Environmental chemistry of soils*. Oxford University Press, New York, 1994. 2

[23] JM Pauwels, E Van Ranst, M Verloo, and A Mvondo Ze. *Manuel de Laboratoire de Pédologie. Méthodes d'Analyses de Sols et de Plantes, Equipement, Gestion de stocks de Verrerie et de Produits chimiques*. Publications Agricoles 28. AGCD et Centre Universitaire de Dschang, Bruxelles, 1992. 2

[24] Edzer J. Pebesma and Roger S. Bivand. Classes and methods for spatial data in R. *R News*, 5(2):9–13, 2005. 130

[25] EJ Pebesma. *gstat User's Manual*. Dept. of Physical Geography, Utrecht University, Utrecht, version 2.3.3 edition, 2001. 129

[26] EJ Pebesma. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7):683–691, 2004.

[27] EJ. Pebesma and CG. Wesseling. Gstat: a program for geostatistical modelling, prediction and simulation. *Computers & Geosciences*, 24(1):17–31, 1998. 129

[28] R Development Core Team. *An Introduction to R*. The R Foundation for Statistical Computing, Vienna, version 2.6.2 (2008-02-08) edition, 2008. 1

[29] BD Ripley. *Spatial statistics*. John Wiley and Sons, New York, 1981. 121

[30] D G Rossiter. *Introduction to the R Project for Statistical Computing for use at ITC*. International Institute for Geo-information Science & Earth Observation (ITC), Enschede (NL), 3rd edition, 2007. 1, 11

[31] GW Snedecor and WG Cochran. *Statistical methods*. Iowa State University Press, Ames, Iowa, 7th edition, 1980. 2

[32] DL Sparks. *Environmental soil chemistry*. Academic Press, San Diego, 1995. 2

[33] P Sprent. *Models in regression and related topics*. Methuen's monographs on applied probability and statistics. Methuen, London,, 1969. 45

[34] RGD Steel, JH Torrie, and DA Dickey. *Principles and procedures of statistics : a biometrical approach*. McGraw-Hill series in probability and statistics. McGraw-Hill, New York, 3rd edition, 1997. 2

[35] WN Venables and BD Ripley. *Modern applied statistics with S*. Springer-Verlag, New York, fourth edition, 2002. 1, 23, 42, 114, 121, 140

[36] R Webster. Is regression what you really want? *Soil Use & Management*, 5 (2):47–53, 1989. 16

[37] R Webster. Regression and functional relations. *European Journal of Soil Science*, 48(3):557–566, 1997. 16, 17, 45

[38] R Webster and MA Oliver. *Statistical methods in soil and land resource survey*. Oxford University Press, Oxford, 1990. 2

[39] DS Wilks. *Statistical methods in the atmospheric sciences: an introduction*. International Geophysics Series 59. Academic Press, New York, 1995. 2

[40] M Yemefack. *Modelling and monitoring soil and land use dynamics within shifting agricultural mosaic systems*. ITC Dissertation 121. ITC Enschede and Utrecht University, Enschede and Utrecht, the Netherlands, 2005. 2

[41] M Yemefack, DG Rossiter, and R Njomgang. Multi-scale characterization of soil variability within an agricultural landscape mosaic system in southern Cameroon. *Geoderma*, 125(1-2):117–143, 2005. 2

Index of R Concepts

+ formula operator, 129
[, 104
~ formula operator, 129

abline, 90
abs, 31, 38
aov, 61
as.numeric, 21
asp graphics argument, 118
attach, 9

biplot, 109
border graphics argument, 11
boxplot, 59
breaks graphics argument, 11
by, 59, 60

car package, 83, 84
center argument, 103
coefficients, 88
col graphics argument, 11, 31
colors, 11
contourplot (package:lattice), 124
coordinates (package:sp), 129, 134
cor.test, 22
correlogram (package:spatial), 121
cov, 22, 69

data.frame, 40
dfbeta, 37
dfbetas, 37
drop argument, 104

eqscplot (package:mass), 124
expand.grid, 134

file.show, 3
fit.variogram (package:gstat), 132
fitted, 30

grid, 118
gridded (package:sp), 134
gstat package, 3, 128, 129, 131, 132

hatvalues, 35
head, 129

identify, 102
influence.measures, 37

krige (package:gstat), 135

lattice package, 3, 124
length, 60
levels, 21
library, 128
lines, 90
lm, 25, 29, 30, 35, 59, 61, 63, 64, 72, 77, 89
load, 5
loess, 90
lowess, 23
lqs, 42

main graphics argument, 11
MASS package, 42, 43, 124

order, 58

pairwise.t.test, 64
par, 19
plot (package:lattice), 131
plot, 11, 118
prcomp, 99, 101, 103
predict, 40, 42

range, 107
rank, 52
read.csv, 4, 5
read.table, 4
reshape, 7
resid, 30
retx argument, 101
row.names, 5
rug, 11

scale, 107
scale argument, 100
sd, 22
search, 128
segments, 31
seq, 11
shapiro.test, 34
solve, 103
sort, 58
sp package, 3, 128, 129, 138, 139
span argument, 90
spatial package, 120, 121
stem, 33, 34
step, 77–79

str, 129
subset argument, 89, 90
summary, 44, 129
surf.ls (package:spatial), 120

t.test, 15
text, 11

update, 86

vif, 83, 86

which, 31

A Derivation of the hat matrix

The "hat" matrix is derived from an interesting way to look at the linear model:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \varepsilon \tag{A1}$$

where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$ are the (unobservable) identically-distributed errors with (unknown) variance σ^2.

Recall that a vector of fitted values $\hat{\mathbf{y}}$ is computed from the design matrix $\mathbf{X}$ and the vector of fitted coefficients $\mathbf{b}$:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} \tag{A2}$$

The "hat" notation, e.g. $\hat{y}$, is used to indicate a fitted (not observed) value; the observed value has no hat, e.g. y.

This can be written separately for each of the n fitted values as:

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in}, \qquad i = 1 \ldots n \tag{A3}$$

Equation A3 shows the expansion of the matrix multiplication of equation A1 by rows. There is one β_j for each of the predictors, including the intercept β_0. So, if we know the coefficients $\mathbf{b}$ we can predict at any value of the predictors, i.e. a given row of $\mathbf{X}$.

In least squares regression, the coefficients $\mathbf{b}$ are solved by least squares estimation from the vector of sample observations $\mathbf{y}$:

$$\begin{aligned}
\mathbf{y} &= \mathbf{X}\mathbf{b} && \text{(A4)}\\
\mathbf{X}'\mathbf{y} &= \mathbf{X}'\mathbf{X}\mathbf{b} && \text{(A5)}\\
(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})\mathbf{b} && \text{(A6)}\\
(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} &= \mathbf{I}\mathbf{b} && \text{(A7)}\\
\mathbf{b} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} && \text{(A8)}
\end{aligned}$$

and substituting into equation A1:

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{A9}$$

so that the "hat" matrix, which is what multiples the observations to get the fits, can be defined as:

$$\mathbf{V} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \tag{A10}$$

that is,

$$\begin{aligned}
\hat{\mathbf{y}} &= \mathbf{V}\mathbf{y} && \text{(A11)}\\
\hat{y}_i &= v_{i1}y_1 + v_{i2}y_2 + \cdots + v_{in}y_n, \qquad i = 1 \ldots n && \text{(A12)}
\end{aligned}$$

The "hat" matrix is so-named because it "puts the hats on" the fitted values: from observed $\mathbf{y}$ to best-fit $\hat{\mathbf{y}}$. Notice that once this is computed, there is no more need

for the predictors in the model matrix $\mathbf{X}$, and there is no need to use the regression equation, i.e., all the information in the regression coefficients vector β. However, the "hat" matrix can only be used to compute the fits $\hat{y}_i$ corresponding to each observation of the target variable y_i, it can not be used to predict at arbitrary values of the predictors; for that, the regression equation is used.

The hat matrix $\mathbf{V}$ gives the weights by which each original observation y_j is multiplied when fitting $\hat{y}_i$. This means that if a high-leverage observation were changed, the fit would change substantially. In terms of vector spaces, the $\mathbf{V}$ matrix gives the *orthogonal projection* of the observation vector $\mathbf{y}$ into the *column space* of the design (model) matrix $\mathbf{X}$.

The overall leverage of each observation y_i is given by its *hat value*, which is the sum of the squared entries of the hat matrix associated with the observation, i.e., column j of the matrix. This is also observation i's entry v_{ii} on the diagonal of the hat matrix, because of the fact that $\mathbf{V} = \mathbf{V}'\mathbf{V} = \mathbf{V^2}$:

$$v_{ii} = \sum_{j=1}^{n} v_{ji}^2 \tag{A13}$$

This is because the "hat" matrix is symmetric.

A.1 Influence of values on prediction

Each row i of the "hat" matrix gives the multipliers of each observation x_j on the predicted value $\hat{y}_i$. The larger the v_{ij}, the more influence observation x_j has on the predicted value $\hat{y}_i$. This raises the interesting question ...

Details of this and many other interesting aspects of the hat matrix are given by Cook and Weisberg [5].