Digital Soil Mapping

Getting Operational

An introductory course of practical exercises for Digital Soil Mapping

February 2011

Prepared by:
Brendan Malone
Digital Soil Mapping (DSM) has matured to become a legitimate branch within the soil science domain. DSM has evolved from a science-driven research phase of the early 1990s to presently, a fully operational and functional process. This evolution is evidenced by the increasing extents of DSM projects from small research areas towards regional, national and even continental extents.

What has been a significant contributor to the evolution of DSM has been the advance in information technologies and computational efficiencies in recent times. Such advances have motivated numerous initiatives around the world to build spatial data infrastructures aiming to facilitate the collection, maintenance, dissemination and use of spatial information. Essentially, fine scaled earth resource information of improving qualities is and/or is gradually coming on-line. This is a boon for the advancement of DSM. More importantly however, the contribution of the DSM community in general to contribute to the development of such generic spatial data infrastructure has been through the ongoing creation and population of regional, continental and worldwide soil databases from existing legacy soil information. Ambitious goals such as those proposed by the GlobalSoilMap.Net consortium whose objective is to generate a fine scale 3D grid of a number of soil properties across the globe provide some guide to where DSM is headed operationally. Furthermore, besides the mapping of soil properties and classes, DSM approaches are also now extended to other soil spatial analysis domains such as those of digital soil assessment (DSA) and digital soil risk assessment (DSRA).

It is an exciting time to be involved in DSM. But with development and an increase in the operational status of DSM there comes a requirement to teach, share and spread the knowledge of DSM. Put more simply, we need to teach more people how to do it and how to do it properly. It is such that this short introductory course of exercises for DSM attempts to share and disseminate some of this knowledge. It is by no means an exhaustive course and is not heavy on the theoretical undertakings of DSM. Rather this course is procedurally orientated and attempts to give those interested, a taste and a conceptual framework to undertake DSM in their own technical fields. This course will introduce you to some basic procedures for handling and manipulating soil and spatial data (which is done frequently and in many environments) and then introduce you to some basic concepts and practices for
building predictive models and ultimately making soil maps. In essence, we want to be able to guide you through the ‘steps’ from legacy soil data (here point observations) to the population of a complete soil information system. Furthermore, legacy soil data also comes in the form of legacy soil maps (created from traditional survey methods) too, and during this course we will also cover some methods for using this information for mapping of soil properties.
A note about this manual and workshop

This manual has been put together in such a fashion that participants are introduced to a sequential process that is always performed for DSM projects. For example, from organising data, both, legacy soil and environmental covariate types, right through to constructing predictive models to populate a soil information system and ultimately the production of digital soil maps. Along the way you will be introduced to the fundamental concepts of DSM which will be presented by verbal means. Each exercise in this manual punctuates these discussions.

For each of the exercises, we will be using a generic soil data set so that everyone can conceptualise for themselves what the outcomes of each exercise is. We have supplied all the materials which include the spatial datasets which complement our generic legacy soil data and a number of softwares that will facilitate each of the exercise outcomes. More importantly each exercise is a step-by-step ‘recipe’ guide so that when things seem a bit complicated or you are not familiar with particular software, the instructions will help you through these issues. Depending on time, it would be worthwhile to have with you, your own data (if available) to carry out each of the exercises. It has been found from these types of workshops that concepts are better understood once you use data for which you are more familiar with.

We hope you enjoy this workshop and get something out of it for which you can use in your own technical fields.
**Digital Soil Mapping: what is it?**

In recent times we have bared witness to the advancement of the computer and information technology ages. With such advances, there have come vast amounts of data and tools in all fields of endeavour. This has motivated numerous initiatives around the world to build spatial data infrastructures aiming to facilitate the collection, maintenance, dissemination and use of spatial information. Soil science potentially contributes to the development of such generic spatial data infrastructure through the ongoing creation of regional, continental and worldwide soil databases, and which are now operational for some uses e.g. land resource assessment and risk evaluation (Lagacherie and McBratney 2007).

Unfortunately the existing soil databases are neither exhaustive enough nor precise enough for promoting an extensive and credible use of the soil information within the spatial data infrastructure that is being developed worldwide. The main reason is that their present capacities only allow the storage of data from conventional soil surveys which are scarce and sporadically available (Lagacherie and McBratney 2007).

The main reason for this lack of soil spatial data is simply that conventional soil survey methods are relatively slow and expensive. Furthermore, we have also witnessed a global reduction in soil science funding that started in the 1980s (Hartemink and McBratney 2008) which has meant a significant scaling back in wide scale soil spatial data collection and/or conventional soil surveying.

To face this situation, it is necessary for the current spatial soil information systems to extend their functionalities from the storage and use of digitised (existing) soil maps to the production of soil maps *ab initio* (Lagacherie and McBratney 2007). This is precisely the aim of Digital Soil Mapping (DSM) which can be defined as “the creation and population of spatial soil information systems by numerical models inferring the spatial and temporal variations of soil types and soil properties from soil observation and knowledge from related environmental variables” (Lagacherie and McBratney 2007).

The concepts and methodologies for DSM were formalised in an extensive review by McBratney et al. (2003). In the paper (McBratney et al. 2003), the *scorpan* approach for predictive modelling (and mapping) of soil was introduced which in itself is rooted in earlier works by Jenny (1941) and Dokuchaev (1883). *SCORPAN* is a mnemonic for factors for prediction of soil attributes: soil, climate, organisms, relief,
parent materials, age, and spatial position. The scorpan approach is formulated by the equation:

\[ S = f(S, C, O, R, P, A, N) + \varepsilon \]

or

\[ S = f(Q) + \varepsilon \]

Long-handed, the equation simply states that the soil type or attribute at an unvisited site \((S)\) can be predicted from a numerical function or model \((f)\) given the factors just described plus the locally varying, spatial dependent residuals \((\varepsilon)\). The \(f(Q)\) part of the formulation is the deterministic component or in other words, the empirical quantitative function linking \(S\) to the scorpan factors (McBratney et al. 2003). The scorpan factors or environmental covariates come in the form of spatially populated digitally available data for instance from digital elevation models and the indices calculated from them (slope, aspect, MRVBF etc); Landsat data and other remote sensing images, radiometric data, geological survey maps, legacy soil maps and data just to name a few. For the residuals \((\varepsilon)\) part of the formulation, we assume there to be some spatial structure. This is for a number of reasons which include that the attributes used in the deterministic component were inadequate, interactions between attributes were not taken into account or the form of \(f()\) was misspecified. Overall this general formulation is called the scorpan kriging method where the kriging component is the process of defining the spatial trend of the residuals (with variograms) and using kriging to estimate the residuals at the unvisited sites.

Without getting into detail with regards to some of the statistical nuances such as bias issues (which can be prevalent when using legacy soil point data for DSM. This is because the sampling has usually been performed with no statistical criteria) that are encountered with using this type of data, the application of scorpan kriging can only be done in extents where there is available soil point data (this is logical). The challenge therefore is what to do in situations where this type of data is not available. In the context of the global soil mapping key soil attributes, this is a problem, but can be overcome with the usage of other sources of legacy soil data such as existing soil maps. It is even more of a problem when neither this information is available either. However, in the context of global soil mapping, Minasny and McBratney (2010) proposed a decision tree structure for actioning DSM on the basis of the nature of
legacy soil data available. This is summarised in the below figure, but bear in mind that this decision tree is not constrained only to DSM at a global scale but at any scale where the user wishes to perform DSM given the availability of soil data for their particular area.

![Decision Tree](image)

**Figure.** A decision tree for digital soil mapping based on legacy soil data (Source: Minasny and McBratney, 2010)

As can be seen from this decision tree, once you have defined an area of interest and assembled a suite of environmental covariates for this area then determined the availability of the soil data there, you follow the respective pathway. SCORPAN kriging is performed exclusively when there is only point data but can be used also when there is both point and map data available. The work flow is quite different when there is only soil map information available. Bear in mind that the quality of the soil map depends on the scale and subsequently variation of soil cover; such that smaller scaled maps e.g. 1:100 000 would be considered better and more detailed than large scaled maps e.g. 1:500 000. The elemental basis for extracting soil properties from legacy soil maps comes from the central and distributional concepts of soil
mapping units. For example, modal soil profile data of soil classes can be used to quickly build soil property maps. Where mapping units consist of more than one component, we can use a spatially weighted means type method i.e. estimation of the soil properties is based on the modal profile of the components and the proportional area of the mapping unit each component covers. As a pre-processing step prior to creating soil attribute maps, it may be necessary to harmonise soil mapping units (in the case of adjacent soil maps) and/or perform some type of disaggregation technique in order to retrieve the map unit component information. Some approaches for doing so have been described in Bui and Moran (2001).

What is the process when there is no soil data available at all? This is obviously quite a difficult situation to confront but a real one at that. While we will not cover this area of DSM in this workshop, the central concept that was discussed by Minasny and McBratney (2010) for addressing these situations is based on the assumed homology of soil forming factors between a reference area and the region of interest for mapping. This research is still in its development, but considering from a global perspective, the sparseness of soil data and limited research funds for new sol survey, application of this Homosoil approach or other analogues will become increasingly important for the operational advancement of DSM.

References


Limited review on software for supporting Digital Soil Mapping

INTRODUCTION

In this short course you will become familiar in the use with a number of licensed and open source softwares and applications. Collectively these tools are some of what are available to digital soil map producers to accomplish their research outcomes. These tools are not DSMcentric and you have more than likely come across them in various other work situations. That being said, the list of tools available now to digital soil mappers is not exhaustive, there are many available and there are some you will probably be more comfortable with. Nevertheless, this course has been structured to use the tools detailed below. During this course you will find yourself jumping from tool to tool in order to fulfil a particular task. It is hoped that rather than becoming versed in a particular set of tools you become familiar with the concepts and logic of DSM after which you can use any tools available to accomplish your work outcomes.

In no particular order, in this course you will predominantly use the following tools:

SAGA GIS

SAGA – an acronym for "System for Automated Geoscientific Analyses" - is a free and open source, cross-platform GIS software. It was originally developed by a small team at the Department of Physical Geography, University of Göttingen, Germany, and is now maintained and extended by an international developer community.

The versatility of SAGA for a range of geo-spatial applications can be measured by its growing availability of geo-scientific methods. These are implemented in various SAGA modules which are bundled in so-called module libraries. These module libraries are accessible to you in different ways: By a Graphical User Interface (SAGA GUI) or by one of the scripting methods. The scripting methods include a Command Line Interface (SAGA CMD) which allows for batch/bash scripting (i.e. coupling of different modules to automate tasks) and a python interface which gives you also direct access to the SAGA API and is thus more flexible. SAGA GIS modules can also be executed from within the statistical data analysis software R in order to integrate statistical and GIS analyses.
SAGA GIS is an effective tool with user friendly GUI that requires only about 10 MB disk space. No installation is needed. You can load raster, vector and tabular data and visualize them in map views, 3D views, diagrams, scatter plots, etc. The GUI provides great flexibility regarding e.g. color tables, layering of datasets and display of attributes. You have full access to all SAGA modules to process your data and examine the results. This includes a lot of scientific methods you will miss in other GIS applications. Further information regarding SAGA GIS see: http://www.saga-gis.org/ for instruction on operating SAGA within the R console see: http://cran.r-project.org/web/packages/RSAGA/index.html

R

R is a language and environment for statistical computing and graphics. It is a GNU project (free) which is similar to the S language and environment which was developed at Bell Laboratories (now AT&T/Lucent). R can be considered as a different implementation of S and was written and released as open source software by Ross Ihaka and Robert Gentleman at University of Auckland in the 1990s. Since 1997 R has been maintained and extended by an international R-core team (around 15 members) plus 1000s of code writers and statisticians.

The term "environment" is intended to characterize R as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software. R is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Moreover, R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, spatial data analysis, clustering, ...) and graphical techniques, and is highly extensible.

Some operational advantages of R include: It is fast and free; a wide availability functions and sets of functions (packages) – 2739 on last count. This means that users can access state-of-the-art methodologies and functions developed by experts and researchers for free. R has an active user community; it is an excellent environment for simulation, programming and computer intensive analyses. R forces you to think about your analyses. There are some disadvantages also: It is not user friendly when you first begin i.e. minimal GUI; there is no commercial support; there is some
limitation when using particularly large data sets. Further information regarding R, visit: http://www.r-project.org/

**Tinn-R**

Tinn stands for the recursive acronym 'Tinn is not Notepad'. It was initially written as a small ASCII file editor primarily intended as a better replacement for the default Notepad.exe program distributed with Windows. With versions, it grew to a fairly complete, yet simple and compact code editor.

The specific version (Tinn-R) contains enhancements to allow syntax highlighting of S language (in *.R, *.r, *.Q or *.q files), that is, the language used by the open source statistical software R (http://www.r-project.org). Tinn-R is a small, free and simple, yet efficient, replacement for the basic code editor provided by Rgui. Tinn-R can be downloaded from: http://sourceforge.net/projects/tinn-r/

**Vesper**

VESPER, an acronym for “Variogram Estimation and Spatial Prediction plus Error” is a PC-Windows program which provides rigorous spatial prediction techniques. The software was developed by the Australian Centre for Precision Agriculture (ACPA). Initially developed for the precision agriculture industry, VESPER is widely used in a range of disciplinary fields where spatial prediction methods are carried out.

Operated from within a user-friendly GUI, the program provides flexibility to calculate global and local variogram models, undertake global and local kriging in either punctual or block form and output the parameters and estimates in an ASCII text format. The program provides control of the semivariogram calculation and choice of models namely, spherical, exponential, Gaussian and linear with sill. Other functionalities included in the software are boundary and grid file generation applications. VESPER allows user defined neighbourhood and prediction-block sizes along with other advanced controls. It also provides a real-time graphical display of the semivariogram modelling and a progress (and final) map of the kriged estimates. Further instruction on VESPER including downloads can be found at: http://sydney.edu.au/agriculture/acpa/software/vesper.shtml or by reading the manual: http://sydney.edu.au/agriculture/acpa/documents/Vesper_1.6_User_Manual.pdf
**FuzMe**

FuzMe is a PC Windows program for calculation of Fuzzy k-means with/without extragrades. The software was developed by the Australian Centre for Precision Agriculture (ACPA) and is written in Fortran and compiled using Compaq Visual Fortran 6.6 under PC Windows environment. The program needs a "control file" which details the parameters for the fuzzy k-means algorithm and a "data file" containing the data.

Other features of the software include options for a range of distance metrics namely, Euclidian, Diagonal, and Mahalanobis. Iterative running of the software with a range of cluster numbers and fuzzy exponent values can also be performed. Other advanced controls include bootstrapping and jack-knifing capabilities. The program is run through an easy to use GUI and outputs are exported in text files. Further instruction and downloads can be found at: [http://sydney.edu.au/agriculture/acpa/software/fuzme.shtml](http://sydney.edu.au/agriculture/acpa/software/fuzme.shtml)

**JMP**

JMP, pronounced ‘jump’ is a commercially available licence statistical software created, developed and released by SAS. JMP is statistical software with a friendly GUI to display and analyse data. Some of the components of JMP includes:

- A spreadsheet for viewing, editing, entering, and manipulating data
- A broad range of graphical and statistical methods for data analysis
- Extensive design of experiments
- Options to select and display subsets of the data
- Data management tools for sorting and combining tables
- A calculator for each table column to compute values
- A way to group data and computing summary statistics
- Special plots, charts, and communication capability for quality improvement techniques
- Tools for moving analysis results between applications and for printing
- A scripting language for saving frequently used routines

Statistics are organized into logical areas with appropriate graphs and tables, which help find patterns in data, identify outlying points, or fit models. Appropriate analyses are defined.
and performed for you, based on the types of variables you have and the roles they play. JMP offers descriptive statistics and simple analyses for beginning statisticians and complex model fitting for advanced researchers. Standard statistical analysis and specialty platforms for design of experiments, statistical quality control, ternary and contour plotting, and survival analysis provide the tools you need to analyse data and see results quickly. JSL coding allows custom analyse to be run within the software environment. For further instruction consult any one of the released manuals: http://www.jmp.com/support/downloads/pdf/jmp_user_guide.pdf.
Digital Soil Mapping:- Methodologies Based on SOIL POINT DATA

Overview

Methodologies for DSM (Global soil mapping) based on legacy data were discussed in Minasny and McBratney (2010). The process can be summarised in the figure below whereby for an area of interest we assemble environmental covariates and existing soil data. The type of soil data available for an area will determine which method to be used for DSM.

For now we will concentrate on the scenario whereby we have only soil point data. It is such with this situation, soil properties can be interpolated and extrapolated across the immediate area of interest using a combination deterministic and stochastic modelling approaches, hence the name *scorpan* (deterministic) kriging (stochastic).

![Figure: A decision tree for digital soil mapping based on legacy data (From: Minasny and McBratney 2010).](image)

The following exercises have been arranged so as to guide you through the steps one takes for generating DSM products using soil point data. For these exercises, demonstration of the techniques will use some of the Edgeroi Dataset (McGarry et al. 1986). Thus our ‘area of interest’ in the following exercises is the area to the north of...
Narrabri (30.32S 149.78E) the lower valley of the Namoi River, approximately 500km NNW of Sydney (see McGarry et al. 1986 more details of this area and the dataset). The spatial coverage of the 341 site locations that soil was collected from and lab analysed means that the areal extent of this study area is approximately 1500lm². At each of the soil sites (mostly), soil was collected from the depth intervals of 0–0.1, 0.1–0.2, 0.3–0.4, 0.7–0.8, 1.2–1.3 and 2.5–2.6m. Some of the data we will be working with will be Clay %, volumetric carbon (TOC), and soil pH.

The work flow of the exercises is:

- In exercise 1, we want to view the site locations in their spatial extent. We will also visualise the soil data in this environment in order to form generalised views soil variability across the area. We will be using Google Earth as our visualisation tool.

- In exercise 2, we begin data preparation for DSM which will entail harmonising this soil profile information to derive values of the soil properties at standard depths. The standard depths we will be using are those listed for the GlobalSoilMap specifications. We will be using mass-preserving splines to perform this harmonisation procedure using a couple of available tools.

- In exercise 3, we will begin looking at some environmental covariate data specific for the study area but also discuss more generally what types of covariates are important for DSM and where and how we might acquire or derive them. Then we look at the process in which we couple or join (intersect) this covariate information with the soil data so as to prepare for the modelling components.

- In exercise 4, we will work through modelling processes for DSM. To start we use simple linear models and then progress to more complicated parametric and non-parametric modelling techniques. We will then introduce regression kriging. Some demonstrations of other modelling techniques will ensue in this exercise. We will also have a look at the processes for mapping soil classes with a DSM framework (using other data of course). Creating various soil attribute maps at the standard depths and soil class maps are the expected outcome of this exercise.

References

Exercise 1: Visualisation of soil data within a GIS

Introduction

Almost everyone will be familiar with one or more GIS platforms for visualising data within a spatial environment. This exercise has dual purposes. First is to introduce you to the ‘area of interest’ and the subsequent data that is available here. We will be using probably the most widely used GIS platform available, Google Earth. Second is to head straight into operational mode for using R. This exercise will give people new to R a basic introduction. To those with some experience or well versed enthusiasts, this is an opportune time to be re-acquainted or explore some of the vast capabilities of R. Nevertheless, this exercise and all subsequent exercises are written under the assumption that everyone in a beginner even though this may not be the case.

Google Earth is a nice GIS to work with as it is free and the images are continually refreshed or updated. While it does have a limited capacity to perform most spatial operations you would use in an ESRI type GIS, it is perfect for visualisation and familiarising oneself with their area of interest. This is especially true for visualising soil data within a geographical context where it is possible to note things such as positional anomalies. It is also a useful platform for planning soil survey.

Creating KML files can be done in many software packages. It can also be done within R using the Rgdal (URL: http://cran.r-project.org/web/packages/rgdal/rgdal.pdf) package which has bindings for the Geospatial Data Abstraction Library created by Frank Warmerdam (http://home.gdal.org/~warmerda/). This is provided that we tell R (import data) what we want to visualise and where in the world this data is located. This additional functionality is made possible by using the spatial data package (sp).

The KML files we will be creating in this exercise start with just simple point vectors of the site locations. Then we will create files decidedly more complex because we are able to visualise not only the site locations of each profile but also all the observations at each depth of each site in a pseudo 3D kind of way. Albeit the observations appear above ground but you can visualise all the available data nonetheless. While we cannot perform any statistical analysis at this stage, visualising the data in this setting provides the opportunity to begin thinking about how the soil varies across the ‘area of interest’ for example comparing the texture grades of the soil profiles.
Outcomes

By the end of this exercise you will have created a number of KML files which will include:

- The site locations of the Edgeroi Dataset
- The observed soil data at each site

Part 1: Display locations of the Edgeroi dataset point locations in Google Earth

- First open TinnR: All Programs>Tinn-R>TinnR
- Now on the toolbar you will see an icon that looks like this 🔄. Click on this icon. This synchronises both softwares so that you are able to send scripts directly from TinnR to R. You should see something on your screen like below.

- Through the File>Open menu of TinnR navigate to the location where the script for this exercise is. The root location may be different for most people but in the case of it being the C Drive, the pathway is
C:\Sydney_DSM2011\Exercises\Display. Click on the file titled visSoilDat.r. The script will now open up in the main window of TinnR.

- First display line numbers if they have not already appeared. This can be toggled by pressing Ctrl+L.

- The script that you see is essentially a set of instructions that we want to send to R so that it can output the files we want. To send these instructions to R there are two main ways: line-by-line or entire chunks of script.

- To start with we will send the instructions line-by-line. First click on the text you see in **Line 2**. It will highlight yellow. This command initialises the working directory to tell R where the files it needs to perform its list of instructions and places its outputs. Therefore you will need to change the root location. At the moment it is set to the **C drive**. The pathway should remain unchanged. Once this is done you want to send this line to R by pressing the send line icon which is located on the toolbar. This line should appear below in the R console in red text (meaning that everything went ok) like below.

  ```r
  > setwd("C:\Sydney_DSM2011\Exercises\Display")
  > |
  ```

- Now send **Line 3** following the same steps. This command loads up a specific (non-native) function which is necessary for creating our specialised KML files.

- One thing you will notice is a lot of green text which indicate comments that are not read by R but act as an instruction for the person using the scripts. All comments are preceded by a # symbol after which you can write anything you like. When writing multiple lines of comments be sure to begin each new line with a # symbol.

- To send chunks of script with one click we first highlight the portion of script we want to send and then click on the icon. Do this by sending **Lines 6-9** to R. As described in the discussion of the various softwares used in this workshop, R has many ‘base’ functions for performing a range statistical tasks. However for our tasks we need some extra functionalities
(that are not part of the base R) for example: handling spatial data, projections and writing KMLs. Therefore we need to load up a number of packages/libraries that are necessary for R to carry out our instructions like for example **rgdal** for handling projections etc and **sp** for manipulating our spatial data. The other 2 libraries also have their necessary functionalities.

- If there was an issue with sending this chunk of script such that R output came out something like this:

  ```
  > Error in source(.trPaths[5], echo = TRUE, max.deparse.length = 150) : object ".trPaths" not found
  ```

  There is something wrong with the way TinnR communicates with R. This is complicated to fix but the short workaround for this is to copy the script below:

  ```r
  .trPaths <- paste(paste(Sys.getenv('APPDATA'), 'Tinn-R\tmp', sep=''),
                 c('', 'search.txt', 'objects.txt', 'file.r', 'selection.r', 'block.r',
                   'lines.r'), sep='')
  ```

  Now paste it at the end of the script that you have open in TinnR. Now send this short instruction (line-by-line) to R. This will have fixed the problem.

- With all these initialisation jobs done we can now move on to the actual tasks of this exercise. Part 1 is the creation of a KML showing where in our ‘area of interest’ the site locations of the Edgeroi dataset are.

- Send **Line 15** to R. This command brings into the console our input text file and names this object **dat_p**. In the R console type **dat_p**. You will see something like this:
The `dat_p` object is a data frame of 3 columns × 341 rows (the number of sites in the Edgeroi dataset). The 1st column is the site id and the 2nd and 3rd columns are their projected spatial coordinates. The projection they are currently in is WGS84 UTM Zone55.

- The projection they are currently in is not the native projection of Google Earth. Thus we need to transform these projected coordinates to geographic whereby the geographic coordinate reference system (CRS) is WGS84. Send Lines 16-18 as a chunk. For these particular commands we are calling upon some of the `sp` library functionality to set which columns or the coordinates (Line 15), indicate what the existing CRS is (Line 17), then indicate that we want to transform the existing CRS to WGS84 (Line 18). As a shorthand, we are specifying each of the CRS by their EPSG (European Petroleum Survey Group) codes. It is also possible to specify the CRS by its Proj4 type code as well. A nice website to find such information on most CRSs is: [http://spatialreference.org/](http://spatialreference.org/)

- Now send Line 19 to R. This command writes the first of our KML files. To open this you can navigate to it through your file folder and double click which should automatically open it up in Google Earth. Otherwise Open Google Earth first and navigate to where you wrote it to through the File>Open menus. If everything is OK you should see something like this. It is now a good time to familiarise yourself with the Edgeroi area:
Part 2:- Display soil attribute data at each location of the Edgeroii dataset in Google Earth

- For Part2 we move on to plotting all the observed data at each location in a 3D type visual representation. Send Line 28 to R. This overwrites the previous dat_p object. Its structure is now this:

```r
> dat_p<-read.table("Edgeroii_Clay.txt", sep="", header=T) # Bring data in
> length(dat_p[,1])
[1] 2175
> dat_p

<table>
<thead>
<tr>
<th>level</th>
<th>Soil.ID</th>
<th>Soil.ID</th>
<th>east</th>
<th>north</th>
<th>Upper Boundary</th>
<th>Lower Boundary</th>
<th>Clay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ed001</td>
<td>741912</td>
<td>667688</td>
<td></td>
<td>8</td>
<td>92.2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>ed001</td>
<td>741912</td>
<td>667586</td>
<td>10</td>
<td>10</td>
<td>56.2</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>ed001</td>
<td>741912</td>
<td>667586</td>
<td>30</td>
<td>40</td>
<td>51.9</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>ed001</td>
<td>741912</td>
<td>667586</td>
<td>70</td>
<td>80</td>
<td>49.7</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>ed001</td>
<td>741912</td>
<td>667586</td>
<td>120</td>
<td>130</td>
<td>55.6</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>ed001</td>
<td>741912</td>
<td>667586</td>
<td>250</td>
<td>280</td>
<td>81.8</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>0</td>
<td>10</td>
<td>61.1</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>10</td>
<td>20</td>
<td>65.2</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>30</td>
<td>40</td>
<td>65.4</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>70</td>
<td>80</td>
<td>65.4</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>120</td>
<td>130</td>
<td>68.8</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>ed002</td>
<td>744662</td>
<td>667796</td>
<td>250</td>
<td>280</td>
<td>56.3</td>
<td>32</td>
</tr>
<tr>
<td>13</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>0</td>
<td>10</td>
<td>66.7</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>10</td>
<td>20</td>
<td>40.3</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>30</td>
<td>40</td>
<td>69.0</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>70</td>
<td>80</td>
<td>71.9</td>
<td>40</td>
</tr>
<tr>
<td>17</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>120</td>
<td>130</td>
<td>76.1</td>
<td>42</td>
</tr>
<tr>
<td>18</td>
<td>ed003</td>
<td>747512</td>
<td>667793</td>
<td>250</td>
<td>280</td>
<td>78.9</td>
<td>44</td>
</tr>
</tbody>
</table>
```
This data frame is 7 columns × 2175 rows where each row is a particular observation at a specified depth at a particular location. Columns 1 and 2 indicate the profile number and id respectively. Columns 3 and 4 are the spatial coordinates. Columns 5 and 6 are the upper and lower depths of observation respectively. Column 7 is the attribute concentration, where in this example it is clay %. As you will observe there are multiple observations at each profile which therefore correspond to the depths of observations. Take note of the structure of this data frame as it should be a template to follow if you wish to repeat this part of the exercise with your own data.

- Send Lines 31-34. This initialises then transforms the CRS to geographic WGS84 as done before.
- Send Lines 38-45. Here we are writing the output from Line 34 to a data frame and then initialising some parameters to use in our specialised plotting function in the way of creating objects out of each of the columns Have a look a the structure of the data and take note as the function can only operate when it is structured and ordered like this:

```r
> dat_p.11<-as.data.frame(dat_p.11)  # see notes at end of routine on specific format required for this object.
> Level<- dat_p.11[,1]  
> ID<- dat_p.11[,2]  
> Upper<- dat_p.11[,3]  
> Lower<- dat_p.11[,4]  
> Target<- dat_p.11[,5]  
> East<- dat_p.11[,6]  
> North<- dat_p.11[,7]  
> dat_p.11
Level ID Soil ID Soil ID Upper Boundary Lower Boundary Clay east north
1 1 e1001 0 0 52.8 169.5079 -30.00916
2 1 e2001 10 23 56.2 169.5079 -30.00461
3 1 e1001 30 60 51.9 169.5079 -30.00461
4 1 e1001 70 60 49.7 169.5079 -30.00461
5 1 e2001 120 110 55.8 169.5079 -30.00461
6 1 e1001 150 160 51.8 169.5079 -30.00461
7 2 e1002 0 10 51.1 169.5079 -30.00471
8 2 e1002 10 20 63.4 169.5079 -30.00471
9 2 e1002 30 40 55.4 169.5079 -30.00471
10 2 e1002 70 80 55.4 169.5079 -30.00471
11 2 e1002 100 110 55.4 169.5079 -30.00471
12 2 e1002 150 160 58.3 169.5079 -30.00471
```

- Send Lines 48-50. Here we set up the object `attribute` which is a KML file that we want to write our data to. Leave the name as is for now but when we wish to visualise other aspects of the data rather than the target variable such as the lower depth it is necessary to change the file name. The `feat` object is the data we want to visualise such that we could create KML of
any one of the objects from **Lines 38-45**. As you will later see, while the data displayed may change, the symbology arrangement will always reflect the target variable concentration. The `dfact` object can be likened to an exaggeration parameter such that increasing it will increase the dimensionality of the data when we visualise it. It is best to leave for now, but you can experiment with it later.

- Now send **Line 53** to R. This is where we call up our specialised plotting function (**donuts**) which takes in the parameters we have just initialised i.e. `dat_p.ll`, `attribute`, `feat` and, `dfact`. This function was adapted from R scripts developed by Tomislav Hengl (Wageningen, The Netherlands).

- If you open up the generated KML in Google Earth and focus in on a site you should see something like this.

Each icon represents a particular observation at a particular depth. The size of the icon and its color is related to the target variable concentration. The color symbology is such that:
The way the donuts function works is like a histogram function where the distribution of the target variable is divided into bins of equal proportions. Which bin each observation belongs to is determined by its observed concentration. Each bin is then given an icon size and color.

At the moment we can just see the target variable concentration. We might also like to see at what depth that observation was observed at. This can be simply done by changing the values of the attribute and feat objects. Such that if we wanted to plot the lower depths of observation Lines sent to R would look like:

```
> feat<- Lower  # Can only either be: Level - ID - Upper - Lower - Target - East - North
> dfact<- 700   # A graphical parameter controlling the length of stake
> do_donut<- donuts(dat_p.11, attribute, feat, dfact)  # run function
```

When back into Google Earth you will now see:
It is up to you to determine whether this type of visualisation is useful. For a first cut exploratory analysis it is not bad and gives you some idea of the data availability for the ‘area of interest’ for which you can begin to form a general mental model of soil variability across this landscape.

You could repeat the steps for other features within this dataset or repeat this part of the exercise by visualising some of the other data available such as pH (Edgeroi_pH_XY.txt) and/or carbon (Edgeroi_Carbon_XY.txt). Just specify the correct filename at Line 28. You could also try this exercise on your own data, but keep in mind the unfortunately strict data structure requirements.

END OF EXERCISE
Exercise 2: Fitting mass preserving splines to soil profile data using R

Introduction

The traditional method of sampling soil involves dividing a soil profile into horizons. The number of horizons and the position of each are generally based on attributes easily observed in the field, such as morphological soil properties (Bishop et al. 1999). From each horizon, a bulk sample is taken and it is assumed to represent the average value for a soil attribute over the depth interval from which it is sampled. There are some issues with this approach, particularly from a pedological perspective and secondly from the difficulty in using this legacy data within a Digital Soil Mapping (DSM) framework where we wish to know the continuous variability of a soil both in the lateral and vertical dimensions. From the pedological perspective soil generally varies continuously with depth; however, representing the soil attribute value as the average over the depth interval of horizons leads to discontinuous or stepped profile representations. Difficulties can arise in situations where one wants to know the value of an attribute at a specified depth. The second issue is regarding DSM and is where we use a database of soil profiles to generate a model of soil variability in the area in which they exist. Because observations at each horizon for each profile will rarely be the same between any two profiles, it then becomes difficult to build a model where predictions are made at a set depth or at standardised depth intervals.

Legacy soil data is too valuable to do away with and thus needs to be moulded to suit the purposes of the map producer, such that one needs to be able to derive a continuous function using the available horizon data as some input. This can be done with many methods including polynomials and exponential decay type depth functions. A superior continuous depth function is the equal-area quadratic spline function. The usage and mathematical expression of this function have been detailed in, Ponce-Hernandez et al. (1986), Bishop et al. (1999) and Malone et al. (2009). A remarkable feature of the spline function is that it is mass preserving or in other words the original data is preserved and can be retrieved again via integration of the continuous spline. No further details of the spline will be discussed as there is a wide literature base in addition to that cited already which discuss the usage of splines.
In this exercise we will use legacy soil data and the spline function to prepare data to be used in a DSM framework. This will specifically entail fitting splines to all the available soil profiles and then through a process of harmonisation, integrate the splines to generate standardised depths of observation. The tools that are available for fitting splines to soil profile data are varied. We will use a couple in this exercise such as the spline tool from CSIRO and scripts that have been written in R. This particular exercise details the workflow for fitting splines using R.

**Outcomes**

By the end of this exercise you will have:

- Become familiar with the parameters required for and processes for fitting splines to soil profile data
- Fitted splines to and saved to text files the outputs of the mass preserving spline function for each of the 3 soil attributes from the Edgeroi dataset.

**Method**

The function for fitting splines to soil profile attribute data is performed within The R programming environment. Keep in mind that this is not an exercise in computer programming. Rather, the work flow of fitting the functions goes from importing the data, then setting up some necessary parameters, then fitting the spline function followed by the production of some simple plots.

To get some familiarity with the R environment and how to input data and input parameters, the exercise will first begin with fitting a spline to a single profile:-

one_profile_C.txt.

- Navigate to the *Exercises* folder in your directories. Here you will find a file called *Run splines.r*. Double click on this and it will open up in the TinnR environment. TinnR is a script editor in which R scripts can be written and edited.

- The first task is to press **Ctrl+Alt+L** which adds the line numbers to the script.

- **Lines 3 and 4** initialise the working directory to *C:\Sydney_DSM2011\Exercises* (you will probably need to change the location of the root directory) and then call up the functions used in this script namely the spline function (*ea_splines*) and a plotting function (*plotAll*). These lines need to be sent to the R console. To do this you must first
synchronise TinnR with R. Do this by pressing the R control Icon. R should then open up. To send Lines 3 and 4 to the R console, first highlight Line 3 then press the R send:line icon. The line will have been sent to R if you see it in red in the R console. Repeat the step again for Line 4. Alternatively send both lines at once by highlighting them both and then click on the send selection icon.

- Line 7 is where we bring the data in. Send line to R. We have named the one_profile_C.txt file as the object dat. Click into R and then type dat. What appears is a 4 column matrix with 8 rows (8x4matrix):

```r
> dat<-as.matrix(read.table("one_profile_C.txt", sep="", header=TRUE))
> dat

   Soil.ID Upper.Boundary Lower.Boundary C.Kg.m3
[1,]    1         0          10  20.7156177
[2,]    1        10          20  11.7077764
[3,]    1        30          40  8.2305018
[4,]    1        50          60  6.2999976
[5,]    1        70          80  2.3998524
[6,]    1       120         130  1.9801815
[7,]    1       250         260  0.6006455
[8,]    1       350         360  1.2343327
```

The first column is the soil profile id (must be numeric 1:x). As we are only fitting a spline to one profile all we see in column 1 are 1s. Therefore as you will probably notice each row is an observation at a particular depth interval. Columns 2 and 3 are the upper and lower depth intervals of each observation respectively. For this profile, observations were made a specific depths rather than at the horizon. In column 4 is the attribute concentration at each depth interval which in this case is organic carbon C (kg/m3). For future work it is critical that this data structure is noted and adhered to when attempting to use this script to fit splines i.e. the 4 column model.

- Send Lines 10-14 to R. These are the parameters of that need to be initialised prior to fitting the splines. For example ndata is an object which tells how many profiles there are; lam is the lambda parameter which controls trade-off between the goodness-of-fit of the spline with the roughness of the fit. Increasing lam results in a coarser fit while decreasing it increases the sensitivity of the spline to change accordingly with the observed data often resulting in a closer and smoother fit. In this exercise we will set lam to 0.1.
Then next parameter is \textbf{mxd} which stipulates the maximum depth which we want the construct the spline to. Here we will set this to \textbf{200}, indicating we want only to predict to 200cm regardless of the observed data which is below this depth. The next object \textbf{d} is probably the most critical because it specifies the new depths at which we want to harmonise all our profile data to. While we are only fitting a spline to 1 profile here, the work flow of the function is that a continuous function is first constructed from which we then integrate from the spline the averages between the standard depth intervals. Here the \textbf{d} object is a vector of the numbers (0, 5, 15, 30, 60, 100, 200). This vector indicates the depth intervals in which we want to extract from the spline ie 0-5cm, 5-15cm, 15-30cm, 30-60cm, 60-100cm, and 100-200cm. These standard depths happen to be the depth specifications agreed on for reporting for the GlobalSoilMap.Net project. However, these standard depths can be changed as such to suit. In this exercise we will not change the values of the \textbf{d} object. The \textbf{s} object is a parameter which is used for estimating the uncertainty of the spline fit, but we will not explore this further in this exercise.

- **Line 17** is where we call up the \textbf{ea_spline} function. As you will notice all the parameters we have initialised as well as the actual data all become inputs. Send this line and it will quickly fit a spline to the data.

- There are 3 objects (lists) that are returned to the \textbf{int_s} object when you run the \textbf{ea_spline} function. We will only be concerned with 2 of them, namely the output for the averages of the spline at the standard depths (\textbf{nyfit}) and the continuous spline to the maximum depth of interpolation (\textbf{spfit}). We will only need to use the \textbf{nyfit} object for future work. The \textbf{spfit} object will only be used for some plotting later on. Send **lines 18-20** to R. Line 20 allows you to save \textbf{nyfit} to a text file. What you will observe when you look at the \textbf{nyfit} object (type \textbf{nyfit} into the R console) is a matrix of 7 columns:

```r
> nyfit<-int_s[[2]]  # Dataframe of the spline averages at the standard sspth
> nyfit

[1,] 21.15563 15.83023 9.806157 7.397733 2.754576 1.711759  380
```
The first 6 columns correspond to the concentration of the soil attribute at each standard depth e.g. 0-5cm, 5-15cm and so on. The last column indicates the maximum depth of the observation for the profile. This is just a carry over from the spline function but can be helpful when the maximum depth of observation is below the maximum depth at which we wish to fit the spline to. In these cases the spline will only be fitted to the maximum depth of observation.

- Plotting provides an aesthetic touch which allows visualising the data and subsequent fits of the spline to this data. To plot our one profile send lines 25-27 to R. In future work when working with multiple profiles you can plot specific profiles or a number of profiles in sequence by altering the numbers in lines 25 and 26. For example to plot all plots line 25 is 1 and line 26 is ndata. If you wanted to plot profiles say 10 to 15, line 25 would be 10 and line 26 would be 15 and so on. By sending all these lines to R a plot will emerge and automatically save to your working directory. What you will see is a basic plot with 3 important pieces of information:
The hollow black polygon lines represent the actual observed profile data at their observed depths. The red line is the continuous spline fitted to this data. While the solid green polygons indicate the averages of the spline at the standard depths. The x axis is the target variable concentration, here OC, while the y-axis is the soil depth.

This concludes the step-wise process for fitting splines to the single profile. It is now time to begin fitting splines to multiple soil profiles and saving the outputs. The files that are available for doing this are Edgeroi_Carbon.txt, Edgeroi_Clay.txt, and Edgeroi_ph.txt which are all in your working directory. All that is required is to follow the above steps again. Please also note the data structure of these data to see how it is set out. In future exercises we will be using the nyfit objects of each attribute so it is wise to save them with the following titles: fitted_carbon.txt, fitted_clay.txt and fitted_ph.txt respectively.
References


END OF EXERCISE
Exercise 3: Data preparation for digital soil mapping

Introduction

By now you will be familiar with some of the environmental covariates that are useful for predicting soil properties and classes within a DSM framework. In this exercise we will be exploring a number of these covariates that have been compiled for the Edgeroi area, which specifically are:

<table>
<thead>
<tr>
<th>Covariate</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation</td>
<td>DEM</td>
</tr>
<tr>
<td>Altitude above channel network (AOCN)</td>
<td>DEM</td>
</tr>
<tr>
<td>Direct sunlight insolation</td>
<td>DEM</td>
</tr>
<tr>
<td>Slope</td>
<td>DEM</td>
</tr>
<tr>
<td>Multi-resolution valley bottom flatness (MRVBF)</td>
<td>DEM</td>
</tr>
<tr>
<td>Mid-slope position</td>
<td>DEM</td>
</tr>
<tr>
<td>Terrain wetness index</td>
<td>DEM</td>
</tr>
<tr>
<td>Band 1</td>
<td>Landsat</td>
</tr>
<tr>
<td>Band 2</td>
<td>Landsat</td>
</tr>
<tr>
<td>Band 3</td>
<td>Landsat</td>
</tr>
<tr>
<td>Band 4</td>
<td>Landsat</td>
</tr>
<tr>
<td>Band 5</td>
<td>Landsat</td>
</tr>
<tr>
<td>Band 7</td>
<td>Landsat</td>
</tr>
<tr>
<td>NDVI (B4-B3)/(B4+B3)</td>
<td>Landsat</td>
</tr>
<tr>
<td>B3/B2</td>
<td>Landsat</td>
</tr>
<tr>
<td>B3/B7</td>
<td>Landsat</td>
</tr>
<tr>
<td>B5/B7</td>
<td>Landsat</td>
</tr>
<tr>
<td>K %</td>
<td>Radiometrics</td>
</tr>
<tr>
<td>Thorium ppm</td>
<td>Radiometrics</td>
</tr>
<tr>
<td>Uranium ppm</td>
<td>Radiometrics</td>
</tr>
</tbody>
</table>

The outcome of this exercise is to prepare both the harmonised soil data (from the previous exercise) and the covariate data into a usable format for modelling. This entails intersecting the soil data with the covariate data at the co-located positions and retrieving the output.

The work flow of this exercise:

- First begins with some visualisation of these covariates within a GIS for which we will be using SAGA GIS and Google Earth.
- Next we then want to format the data that was outputted in the previous exercise (the nyfit objects for each of the soil attributes) to vector type.
- We will then use the point shapefile to intersect with the available covariates.
- Export files: soil data with appended covariate data (points) and tabulated covariate data for entire study area

**Objectives**

Thus there are two main objectives of this exercise:

- Export table of harmonised soil data (3 attributes) with intersect covariate data
- Export table of covariates for the entire Edgeroi area

**Method**

In this exercise we will use SAGA GIS, Google Earth, R & TinnR. And JMP

- First open up SAGA. Depending on the root directory the saga_gui.exe is located by navigating to the **:\Sydney_DSM2011\Software\SAGA-GIS folder.

- While SAGA has an extensive import and export compatibility with numerous grid file formats, in this exercise all the grid files we will be using are in the native SAGA grid file format (.sgrd). To open the 20 covariate files for the Edgeroi, go to File>>Load Grid menu and navigate to the **:\Sydney_DSM2011\Edgeroeri_SAGA_grids folder. Highlight all the grid files (20) and click open.
In the **Workspace** window of the **Data** tab you will see this type of display:

What you will see are the names of the 20 covariates. At the top of this list is information relating to the pixel resolution (90m), the number of rows and columns of each grid and their spatial extent.

- Viewing any of the grids in the **Map window** can be simply performed by double clicking on the grid name in the **Workspace window**. A tool bar will become visible which will allow you to zoom in and pan around the grid etc. You can also view multiple grids in the **Map window** and synchronise the viewing by going to the **synchronise map extents** option in the map menu.
Method for creating image overlays in Google Earth with SAGA

This short section will introduce new users to SAGA GIS to the extensive array of modules available for a wide range of geo-spatial analyses. Here we will create image overlays of some of the environmental covariates for the Edgeroi area to view in Google Earth. It is not necessary that you follow this section but at least it will give you some familiarity with SAGA for future reference.

- First you will need to reproject the covariates from the existing WGS84 UTM Zone 55 projection the WGS84 geographic projection. Go to the modules tab at the bottom of the Workspace window. Navigate to the Projection- Proj. 4 library and double click on the Proj. 4(Command Line Arguments, List of Grids) module.
A new window will open which requires a number of parameters. First are the **source projection parameters** which allows you to define the existing CRS of the covariate data (in Proj. 4 format). The Proj. 4 code for WGS84 UTM Zone 55 is, for which you will need to copy into the empty field:

```
+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84 +units=m +no_defs
```

Next you need to indicate which grids you want to re-project. The field next to the **source** parameter has small icon which when pressed will open up a new parameter window. Click on the >> icon which will move all the covariates across to the empty window. These are all the grids we want to reproject, and then click on Okay.
Next define the Proj. 4 code for the CRS of our target grid ie to Geographic WGS84, which is:

+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

For the interpolation parameter set to Nearest Neighbour. For the target parameter set to user defined. Click Ok

- A new window will open which indicate a number of important parameters for the new target grids CRS including the extents, cell size and row and column numbers. None of these needs changing so just press Okay.
• If all went accordingly, a whole new set of grids will have been created once you look back into the **Workspace** menu under the **Data** tab. Notice these grids are segregated from the original grids. This is a nice feature of SAGA as it avoids confusion when working with multiple grids with different spatial parameters.

• Now is time to create the image and KML files for the covariates. Click on the **modules** tab at the bottom of the **Workspace** window. Locate the **Import/Export Images** Library and then double click on the **Export Image** module. A new window will open where a couple of inputs are required.
Set the **Grid System** to the Geographic CRS. Only one image at a time can be created. Pick one grid for the **Grid parameter** and select this same grid for the **Shade parameter**. It is actually possible to change the colour scheme of the image but this will not be covered here. For the **Image File** parameter you need to set the name and location of the image file you are creating. Be sure to set the format of the image to **.png** type. You can now view this particular covariate in Google Earth.

- Overlaying some of the covariate information with some of the KML files created in **exercise 1** might give you some visual cues on how the soil attributes vary in particular parts of the landscape.

**Method for creating point shapefiles from text data files.**

Back to the job of preparing the sol data for DSM, we will initially use R to quickly convert the soil data outputted from Exercise 3 to shapefile format.

- Open up TinnR and then synchronise with the R console by clicking the icon.
- Through the **file>open** menu navigate to the **\Sydney_DSMM2011\Exercises** folder and select the **SpatialiseSoilDat.r** files.
- By now you will be familiar with the process for send lines of script to the R console from TinnR. Press **Ctrl+L** to bring up the line numbers. Send **Lines 2 and 3** to R. This loads up the **rgdal** and **sp** packages as we need to utilise some of their functionality i.e. initialise the CRS and write shapefiles.
- Send **Line 6** which sets up the directory where we will export the shapefiles to. Remember to change the root directory if required.
- **Lines 6 to 14** are bringing in to the R console the files (**nyfit** objects) that were completed in exercise 2. We are also attaching column names to these files on import. Send these lines to R.
- For both the carbon and clay data for the Edgeroi there are 341 profile observations. For soil pH there are 331. Therefore we will need to write 2
shapefiles. Send **Lines 18-19** to R. These are the UTM spatial coordinates for the each of the profiles.

- Send **Lines 22-23** to R. Here we are making two data frames. To each we are appending the spatial coordinates to the soil data. The first data frame object `soil_dat341` combines both the carbon and clay data. While the object `soil_dat331` just has the ph data and should look like this:

```r
> soil_dat331 <- cbind(c_dat, ph_dat[,1:6])
> soil_dat331
   id      east     north    ph1    ph2    ph3    ph4  ph5    ph6
2 ed003 744663 6679983 7.397981 7.973912 8.825389 8.986887 8.857880 8.773213
3 ed005 747412 6677993 0.147554 0.643590 9.141166 6.977771 0.191990 0.054290
4 ed004 750212 6677033 8.574545 8.942486 9.296729 9.239807 0.039983 0.029050
5 ed005 755212 6676563 8.051859 8.164481 8.770507 8.995035 0.043699 0.059914
6 ed006 755712 6677093 8.527120 8.977679 9.310990 9.429163 0.279270 0.059904
7 ed007 755512 6677873 6.748220 7.570488 8.841277 8.752827 0.050513 0.046002
8 ed008 760312 6676693 7.292016 7.806544 8.502194 8.659362 0.115800 0.613533
9 ed009 766013 6677883 8.208317 8.854577 9.229749 9.210992 8.002348 0.651808
10 ed010 766012 6677403 8.177275 8.012000 9.145076 8.064027 0.002571 0.044899
11 ed011 766612 6677383 7.900498 8.074326 8.210763 8.656151 0.229375 0.911975
12 ed012 772312 6677233 8.518112 8.995808 9.210645 9.238474 0.092053 0.017411
13 ed013 775012 6677233 6.537446 6.564830 6.717718 6.920380 0.094455 0.063217
```

- Send **Lines 28-37** to R. Here we are writing our 2 shapefiles. The sequence for this is that first we define what columns are the spatial coordinates. The we define the CRS (which is WGS84 UTM Zone 55) using the EPSG coding type. Lastly we create the shapefile.

- This concludes the our work in R. Now is time to switch back to SAGA GIS. Bring in the shapefiles to the **SAGA Workspace** by going through the **File>Shapes>load shapes** menus. Navigate to the folder where you saved the shapefiles **Sydney_DSM2011\Exercises** and open both `soil_dat341.shp` and `soil_dat331.shp`. These will appear now in the **Workspace** menu.
You might want to overlay these shapes onto one of the grids. Double click on a grid to display it in the Map window. Then double click on 1 of the two shapes. Another window will appear giving you an option to add this layer to map in the Map window. Select this and click ok.
The task now is to intersect the soil observations with the covariates we have available for the Edgeroi Area. Click on the Modules tab of the Workspace window. Locate the Shapes-Grid library and then double click on the Add Grid Values to Points module. A new window will open that will require a number of parameters.

Into the points field select one of your shapefiles. Ensure that the Results field is set to create. This will create a new temporary shapefile. For the Grids field click on the small icon which will open up a new parameter window. Click on the >> icon which will move all the covariates across to the empty window, and then click on Okay. For the Interpolation method set this to nearest neighbour. This interpolation method will in effect extract the pixel values for which it is ‘sitting’ directly on top of. Click OK. Repeat this again for the other shapefile.

You have now created 2 new temporary files. The see that everything went ok, click on one of the new files and then over in the Object Properties window click on the description tab. Here you will see the summary attribute table of this new temporary shapefile which includes the soil data and the covariate data together at each location.
We then want to export these files so that they can easily be opened up by any software or program that reads text files. The method for doing this is to export our new temporary shapefiles to XYZ format. This is done by going to the Modules tab in the Workspace menu, locating the Import/Export Shapes library where you will find the Export Shapes to XYZ module. A new window will open.

### Table Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Type</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>string</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>O1</td>
<td>byte floating point number</td>
<td>0.004129</td>
<td>0.633882</td>
<td>0.139869</td>
<td>0.032295</td>
</tr>
<tr>
<td>3</td>
<td>O2</td>
<td>byte floating point number</td>
<td>1.472849</td>
<td>0.326390</td>
<td>0.620950</td>
<td>0.017869</td>
</tr>
<tr>
<td>4</td>
<td>O3</td>
<td>byte floating point number</td>
<td>0.197315</td>
<td>0.787386</td>
<td>0.262605</td>
<td>0.018736</td>
</tr>
<tr>
<td>5</td>
<td>O4</td>
<td>byte floating point number</td>
<td>1.715750</td>
<td>0.767350</td>
<td>0.718989</td>
<td>0.017869</td>
</tr>
<tr>
<td>6</td>
<td>O5</td>
<td>byte floating point number</td>
<td>0.198627</td>
<td>0.767350</td>
<td>0.139869</td>
<td>0.032295</td>
</tr>
<tr>
<td>7</td>
<td>O6</td>
<td>byte floating point number</td>
<td>0.474709</td>
<td>0.767350</td>
<td>0.474709</td>
<td>0.017869</td>
</tr>
<tr>
<td>8</td>
<td>CL1</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>9</td>
<td>CL2</td>
<td>byte floating point number</td>
<td>0.264960</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>10</td>
<td>CL3</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>11</td>
<td>CL4</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>12</td>
<td>CL5</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>13</td>
<td>CL6</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>14</td>
<td>sodium</td>
<td>byte floating point number</td>
<td>0.462226</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>15</td>
<td>altitude_ACN</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>16</td>
<td>band1</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>17</td>
<td>band2</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>18</td>
<td>band3</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>19</td>
<td>band4</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>20</td>
<td>band5</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>21</td>
<td>band6</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>22</td>
<td>Direct_Insolation</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>23</td>
<td>elevation</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>24</td>
<td>NodeType_Point</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>25</td>
<td>nh3</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>26</td>
<td>potasuim</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>27</td>
<td>G2</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>28</td>
<td>G7</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>29</td>
<td>Si</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>30</td>
<td>Bt</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>31</td>
<td>ETOE</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>32</td>
<td>Fh</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
<tr>
<td>33</td>
<td>EHI</td>
<td>byte floating point number</td>
<td>0.354286</td>
<td>0.226030</td>
<td>0.196485</td>
<td>0.104485</td>
</tr>
</tbody>
</table>
In the shapes field selected one of you new temporary shapefiles. Tick the Save All Attributes field as well as the Save Table Header. When you click on the File field a new window will open. You will want to save this file to the **:\Sydney_DSM2011\Exercises folder and be sure to save the file type as text type. Click save and then click Ok. Repeat this again for the other shapefile.

- We will look at these new text files in JMP a little later. Firstly we now need to export all the gridded covariate data also to text. This data is required to be in this format once we begin interpolating model parameters from calibration models to every grid location across the study area.

- Locate the Export Grid to XYZ module found within the Import/Export-Grids. A new window will open up. Set the Grid System to the UTM CRS ie the one that says 90; 577x 400y; 738743.558x 6643852.81y. In the Grids field you want to ensure you select all the available grids once the new Grids window opens up. Again we want the save this gridded information to the **:\Sydney_DSM2011\Exercises folder and as text type. Be sure to tick the boxes for Write Field Names and Exclude NoData Cells. Click OK.

- To finish off this exercise it will be useful to have a quick look at these new tabulated datasets. This will be done using the JMP statistical software.

- Open JMP- by locating it through the Start menu.
The fastest way to bring text files into JMP is using the drag and drop method. Firstly we need to adjust some import preferences. In JMP go to File>Preferences then scroll down to Text Data Files. Tick the box for Table contains column headers. Click OK

Now going through to My Computer, navigate to the location of the **:\Sydney_DSM2011\Exercises folder. Highlight the text files that you have just created and then drag them into the grey window of the JMP GUI. Everything should be fine and look like this:

END OF EXERCISE
Exercise 4: Operational Digital Soil Mapping: Modelling

Introduction

This exercise will be one of the most intensive for the course and it will take on many forms or cases. Predominantly we will be working with the data we have been investigating in Exercises 1-3, for the Edgeroi area. During this exercise we will also take a look at some of the functions for DSM of soil classes. For these case studies we will be working with some data from both Tasmania and Western Australia. We will also overview a small subset of models for populating soil information systems. The number of models available to operational digital soil mappers is not exhaustible. The models we will be over-viewing range from simple multivariate linear models for which we will also incorporate regression kriging. Also on the agenda we will look at neural networks in addition to some data mining procedures by using a couple of Rulequest products namely See5 (for predicting soil classes) and Cubist (for predicting soil attributes).

In this exercise we will be working predominantly with JMP i.e. for multivariate linear modelling, neural networks and discriminant analysis for soil classes (Tasmania). We will use VESPER during the regression kriging component. We will also use demonstration versions of the RuleQuest products See5 and Cubist to demonstrate their applicability for DSM. While these are only demonstration versions, they have complete software functionality but are limited in that we can only use small datasets. Nevertheless you will become familiar with the data requirements for these softwares and be able to interpret their outputs.

Outcomes

At the end of this exercise you will have:

- Finalised preparations of the soil and covariate data from the Edgeroi for DSM
- Built simple multivariate linear models of the available soil attributes and a number of standard depths and mapped the functions of these models across the whole area (Edgeroi).
- Become familiar with and adept at the usage of regression kriging for DSM (Edgeroi).
• Performed the same procedures above yet used neural networks instead of linear models for the deterministic component of the *scorpan* approach (Edgeroi).
• Performed a discriminant analysis procedure for predicting and then ultimately mapping soil classes (Tasmania).
• Become familiar with the usage of the See5 software for prediction of soil classes (WA).
• Become familiar with the usage of the Cubist software for prediction of soil attributes (Edgeroi).

**Methods**

The style of learning in this exercise will take a demonstration like form whereby instruction is given ‘live’ on the demonstration computer and projector. You will only need to follow the directions of the demonstrator. This style is in place because there are a number of nuances that are not really important in the context of DSM but rather technical nuances relating to operating the software. You will quickly become fluent in these nuances with more exposure to these softwares.

**Data Mining**

To start up either one of the demo versions of Cubist or See5, their respective folders can be found in the **:\Sydney_DSM2011\Software** folder. Each of these softwares should just run straight from the executable file.

**Cubist**

In this component of the exercise you will be introduced to some data mining tools that are quite popular for DSM. Both have been developed by the same company/person: - RuleQuest/Ross Quinlan. Cubist is a numerical rule-based classifier that is not parametric in the sense that multi-linear models are. Cubist is parametric in the sense that each rule specifies the conditions under which an associated multivariate linear sub-model should be used. There are some accompanying notes of a case study where Cubist was used for DSM. In this demonstration we will work through predicting SOC across the Edgeroi at the 0-5cm depth interval.
**Step 1.** Familiarise yourself with the data structures required by Cubist

**Step 2.** Construct a predictive classifier model for SOC

**Step 3.** Unfortunately for this component we are using only the demo version of this software. As a consequence we will not be able to generate a map. Some time will be spent discussing the outputs given by Cubist which may reference some other projects undertaken to improve clarity.

**Simple Multivariate Linear Modelling**

This component of the exercise is run entirely with the JMP software. Rudimentary maps of outputs can be generated in JMP. It is entirely up to you whether you want to export to a GIS software of choice to make higher quality maps. Instruction can be given if this is wanted.

*Requirements:* Data from Exercise 3 i.e. soil data with joined covariates and table of Edgeroi Covariates (all).

**Step 1.** Open up data in JMP and save tables as JMP native files.

**Step 2.** Filter out the No Data values.

**Step 3.** Construct predictive models of the available soil attributes at the standard depths and save model formulas.

**Step 4.** Build soil maps based on the covariates and predictive soil functions.

**Regression Kriging**

In this component of the exercise we will make use of the prediction functions from the multivariate regression analysis and the subsequent residuals. For this we will use JMP and VESPER (for kriging).

**Step 1.** Derive the model residuals from the predictive functions already calculated

**Step 2.** Discussion of the use of variograms for estimating spatial structure with respect to other methods.

**Step 3.** Perform kriging using localised variogram fits of the residual data onto the regular grid of the Edgeroi area i.e. where you only have information about the covariates. Save the estimated residual residuals

**Step 4.** Build maps of the final predictions i.e. sum of the deterministic $f(Q)$ and stochastic $(\varepsilon)$ components.
**Neural Networks**

This component of the exercise is performed almost exclusively within JMP. It is optional whether you want to investigate and estimate the spatial structure of the residuals.

**Step 1.** Fit successive neural networks that predict simultaneously the soil observations at each standard depth (all soil attributes).

**Step 2.** Determine what the best structure of the predictive models are for each soil attribute, weighing up the relative advantages and disadvantages of model fit quality vs. validation fit quality.

**Step 3.** Build soil maps based on the covariates and the optimal predictive soil functions for each attribute.

**Prediction of Soil Classes:**

**Discriminant Analysis**

In this component of the exercise we will look at DSM of Soil Classes. This component is a short case study of some work done recent in the Meander Valley region near Launceston, Tasmania. See the accompanying notes on this case study. This component will be exclusively carried out with the JMP software. The data set we will use in the component is in the **:Sydney_DSM2011\Exercises\Modelling_data** folder, under the file name **Meander_Tas_Discrim.jmp**. The form of the predictive function we will be looking at for this component is discriminant analysis.

**Step 1.** For some context, read the accompanying notes on why prediction of soil classes was required in this study are

**Step 2.** Construct a predictive model of soil classes based on the available observations and covariates

**Step 3.** Build a soil map based on the covariates that cover the whole area and the predictive function

**Data Mining**
To start up either one of the demo versions of Cubist or See5, their respective folders can be found in the **:\Sydney_DSM2011\Software folder. Each of these softwares should just run straight from the executable file.

*See5*

See5 is a classification software where classifiers are in the form of decision trees and rule sets. For this component of the exercise we will be working with some data from the Dumbleyung area, Western Australia. The data that we will be using is soil class information (WA Soil Super Groups). We will attempt to construct a classifier based on a number of covariates (see accompanying notes) in order to map these classes across the Dumbleyung area.

**Step 1.** Familiarise yourself with the data structures required by See5 (pretty much the same as for Cubist)

**Step 2.** Construct a predictive classifier model for predicting Soil Super Classes (WA)

**Step 3.** Discussion of the output. No maps can be made.
A note on soil class prediction in the Meander Valley Irrigation District.

Introduction

As part of a wider reaching research and policy implementation effort. High resolution digital soil mapping is and will be conducted in the Meander Valley Irrigation District, Tasmania. This area covers approximately 20 000ha and is an agriculturally diverse area with many competing enterprises. While there is legacy soil data for this area, it is limited in spatial coverage density and what data there is, it does not fulfil a lot of the product outcomes which the wider ranging project wishes to meet. For example, a major component of this legacy soil data is in the form of the Quamby 1:100 000 reconnaissance soil map sheet (Nichols 1959) and accompanying report (Spanswick and Zund 1999). In addition to this information not being sufficient for the project outcomes, another issue is that there is incomplete coverage across the Meander Valley and is not at the required spatial resolution (see figure below). Nevertheless, this information is quite valuable, with the reasons why discussed a little further on.

Therefore to get things rolling a sampling effort is needed; essentially to acquire new data, and to get a good spatial coverage of sample sites across the area in order to generate suitable DSM products.

Figure. Clipped section of the Quamby 1:100000 soil map sheet that partially covers the Meander Valley Irrigation District (black polygon).
To implement such a sampling effort, an appropriate sampling design was necessary. It was decided that the Conditioned Latin Hypercube (cLHC) design (Minasny and McBratney 2006) would be the most appropriate. cLHC sampling is a stratified random procedure that provides an efficient way of sampling variables from their multivariate distributions. It provides a full coverage of the range of each variable by maximally stratifying the marginal distribution (Minasny and McBratney 2006). The multivariate distributions in the context of DSM mean the environmental covariates or scorpan factors.

In terms of availability of environmental covariates for the area, there are many. These included a DEM, radiometric data (incomplete coverage), geological map, NIR spectral data products and of course the existing soil map. The existing soil map is too valuable an information source to neglect as it represents the ‘best’ existing source of information regarding the soils in this area (albeit with incomplete coverage). We wanted to ensure that sample sites were allocated to all soil regions in the Meander. The idea then was to correct the incompleteness of this data by filling in the voided area using the existing soil information, expert judgement and DSM procedures.

In this component of the exercise we will attempt, using discriminant analysis, to create a new complete soil map of the Meander. The data available for this are a number of sites for which an expert (Darren Kidd- DPIPWE Tasmania) based on knowledge of the soils in the area was entirely confident that a soil of a particular class existed. There were over 100 such observations which were picked out through a desktop procedure combining GIS software, aerial images and the available covariate data.

With this small database in combination with the other covariate data we can build a predictive model of the soil classes across the Meander. In this little exercise we will be doing a scaled back version of the real thing which was little more complicated because we used a different form of model and that to deal with the incomplete coverage of radiometric data we had to essentially derive two separate predictive models. In this example we will be predicting where there is complete coverage of radiometric and DEM information.
The following is just an illustrative summary of the real work we did for this exercise. It might be good to compare your predicted map with the one we produced.

Existing Soil Map

[Map image]

Expert selection of ‘Training Sites’
Total number of Training sites: 142

Modelling

Model form: classification trees (See5)
Covariates used for modelling. These are the attributes of greatest usage in the classification trees:
- Elevation
- TWI
- Normalised Height
- K%
- Slope
- Thorium ppm
- Valley Depth
- Radiometric Dose_f
- MRVBF

Predicted Soil Map
A note on soil class prediction across the Dumbleyung region, Western Australia

Introduction

The Dumbleyung ‘study area’ is approximately 220km south east of Perth, WA and covers approximately 20000Km$^2$. The soil set for this area is quite well organised and comprehensive. At the University of Sydney we are using sections of this dataset within a couple of DSM related research projects. There are over 3000+ individual soil profiles in this dataset (see figure). For a lot of these profiles, soil physical chemical and data have been recorded at horizon and/or fixed depths. Pretty much every profile has been classified according to the Australian Soil Classification (ASC) (Isbell, 1996) and Western Australian Soil Classification Systems (Soil Super Group Level) (Schoknecht, 1999). A significant proportion of sites have been classified down to the Soil Group Level (WASC).

Figure: The Dumbleyung area and sites of soil observations

In this component of the exercise we will be attempting to use this soil class data (Soil Super Groups) to define a predictive model of where each soil type is likely to occur based on a suite of environmental covariates and then parse these learned rules to where only the covariate information is available.
Methodology

While there are over 3000+ observations, the scaled-down demo version of See5 will only allow us to use around 400 training cases. This however will be enough to conceptualise the partitional process of classification trees for predicting soil classes.

Covariates compiled for this project and the ones of greatest significance (from previous work) in prediction (highlighted in bold):

<table>
<thead>
<tr>
<th>Digital Elevation Model</th>
<th>Landsat data</th>
<th>Radiometric data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevation</td>
<td>Fractional Vegetation Cover</td>
<td>Potassium (%) (3)</td>
</tr>
<tr>
<td>Hill shading (10)</td>
<td>Normalised difference b5/b2 (8)</td>
<td>Thorium (ppm) (2)</td>
</tr>
<tr>
<td>Mid-slope Position</td>
<td>Normalised difference b5/b7</td>
<td>Uranium (ppm)</td>
</tr>
<tr>
<td>MRVBF (1)</td>
<td>Ratio b3/b7</td>
<td>Total count (9)</td>
</tr>
<tr>
<td>Plan Curvature</td>
<td>Ratio b3/b2 (7)</td>
<td></td>
</tr>
<tr>
<td>Profile Curvature</td>
<td>Ratio b5/b7</td>
<td></td>
</tr>
<tr>
<td>Terrain Wetness Index</td>
<td>Soil adjusted vegetation index</td>
<td></td>
</tr>
<tr>
<td>(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slope Height (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incoming solar radiation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standardised Height (5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valley Depth</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A note on using Cubist for DSM

Introduction

Cubist is an application for building rule-based predictive models where the outputs are quantitative values. Cubist models generally give better results than those produced by simple techniques such as multivariate linear regression, while also being easier to understand than neural networks. Its application for DSM is growing such that the rule-based structure of the predictions are likened to the process for building mental models of soil distribution within across a soil landscape. A soil mapper is quite aware that in different parts of the soil landscape, the contribution of the soil forming factors to describe the soil variability differs markedly. These landscape specific nuances are the fundamental element of the mental model that the soil mapper recounts when producing a map. With the rule-based approach of Cubist each rule specifies the conditions under which an associated multivariate linear sub-model should be used. Put simply, you could say that in a particular part of the landscape with defined characteristics you would apply the specific predictive model for this area. In another part of the landscape with different characteristics, a different predictive model is used. It is with this approach that it becomes clear we have quantitative models where in different parts of the landscape the contribution of the soil forming factors to the current state of soil variability varies.

This can be illustrated by observing some output of Cubist for predicting the variability of SOC (0-30cm) across the Edgeroi.


---------------------

Options:

        Permit extrapolation of <=5%
        Maximum of 2 rules

Target attribute `predict`

Read 296 cases (27 attributes) from Edgeroi.data
Model:

Rule 1: [149 cases, mean 11.374192, range 2.642179 to 26.13826, est err 2.335003]
if
elevation <= 224.0509
then
predict = 0.686964 + 0.028 elevation - 0.73 twi + 10.3 r32 + 5.8 r57
+ 4.3 normal_H + 2.4 ndvi + 1.3 mid_slop_pos

Rule 2: [147 cases, mean 17.174997, range 3.777576 to 40.9002, est err 5.148121]
if
elevation > 224.0509
then
predict = -108.084834 + 0.145 elevation - 11.5 r37 + 44.4 normal_H
+ 3.42 twi + 13.9 mid_slop_pos + 7.5 r57 - 0.96 thorium
+ 2.6 potassium

Evaluation on training data (296 cases):

| Average  | error | 3.375600 |
| Relative | error | 0.69     |
| Correlation coefficient | 0.68     |

Attribute usage:

<table>
<thead>
<tr>
<th>Conds</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>100% elevation</td>
</tr>
<tr>
<td>100%</td>
<td>normal_H</td>
</tr>
<tr>
<td>100%</td>
<td>r57</td>
</tr>
<tr>
<td>100%</td>
<td>mid_slop_pos</td>
</tr>
<tr>
<td>100%</td>
<td>twi</td>
</tr>
</tbody>
</table>
Evaluation on test data (40 cases):

<table>
<thead>
<tr>
<th>Average</th>
<th>error</th>
<th>3.960357</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative</td>
<td>error</td>
<td>0.89</td>
</tr>
<tr>
<td>Correlation coefficient</td>
<td>0.34</td>
<td></td>
</tr>
</tbody>
</table>

Time: 0.2 secs

From this example output we can see that there are 2 rules where the only partitioning condition is elevation. It is such that if the elevation is less than or equal to 224m we would apply the first model, otherwise we apply the second model. From some knowledge of the Edgeroi area, there is quite some logic in these rule sets. The first rule is basically being applied in the broad acre cropping area which is located to the west of this study area. The covariates for predicting carbon here include a number of those related to landuse which have been approximated from Landsat data, specifically, NDVI and band ratios b3/b2 and b5/b7. There are also some topographical influences. Radiometric data in this area has no bearing on the predictive model here. However, for the second rule radiometric data in the form of thorium and potassium are important covariates for prediction. As a generalisation we could infer that carbon storage in the west in mainly controlled by landuse factors but in the area to the east which is a more natural vegetation inhabited as well as extensive pasture grazing type land, the control factors of carbon variability are mainly the more static variables relating to parent materials and topographical and hydrological parameters.
References


END OF EXERCISE
Digital Soil Mapping:- Methodologies Based on Legacy Data: DETAILED SOIL MAPS WITH LEGENDS

Overview

In the previous exercise we worked exclusively with soil point data to create digital soil property maps using the *scorpan* kriging approach. We now move onto the broad area of working with conventional soil maps within a digital soil mapping framework.
Exercise 5: working with soil polygon maps for digital soil mapping: deriving soil properties, soil map improvement, disaggregation

Introduction

Conventional soil maps are a valuable source of legacy information that in many parts, represent the only account of soil variability across the defined region. A lot of expert knowledge and time goes into the production of these maps which are the product of systematic field observations complimented with laboratory analysis and analogue cartographic methods. Most of the traditional soil maps are polygon maps depicting the spatial patterning of soil map units. The soil map unit descriptions which accompany the map describe in words the soil-landscape relationships and the likely repeating characteristics of the soil types which exist in the particular map units (Bui and Moran 2001). Hartemink et al. (2010) detail that such traditional maps have had a long history of being used for generalised land use planning and management. However, the authors also detail some of their drawbacks which include:

- They are static as apposed to dynamic thus not amenable for temporal studies.
- They are not amenable for quantitative studies,
- The scale that they are presented at, is often not suitable for a given task or project requiring soil information
- Information is lost. This is because the soil map is a highly summarised account or the collective knowledge of the soil surveyor, field information and other data.
- There is difficulty with integration with other natural resource information that is grid-based for example digital elevation models and satellite imagery.

With improvements in computational and information technologies witnessed in recent times, the availability of digital earth resource information in grid format is growing and is now becoming the norm for spatial information delivery. Couple this with a growing need in the earth resource management and monitoring domains for spatially populated, quantitative soil information (that is finely resolved); it becomes clear that methodologies for handling and the transference of this legacy soil data format into a quantitative format is necessary. Therefore it is not a digitisation effort
required here (although this is quite popularised). What is required is the population of digital soil information systems through the dismantling (or unpacking) of the distilled expert knowledge of the map and accompanying reports.

In this exercise, we will cover some methods and examples for creating soil attribute maps from legacy soil map information. Fundamental to this process is the modal profiles of the components for each mapping unit.

However, this step if you like could be considered the last step of the ‘dismantling’ process because often the soil map needs to be updated and/or disaggregated into their component units first. By the meaning of ‘updated’, this approach is a means of changing the boundary conditions of the mapping units i.e. away from a crisply delineated model to a more continuous or boundless representation of soil attribute variability which can be facilitated with the use of ancillary digital information.

Harmonisation of adjacent soil maps is also an important pre-processing tool. Often adjacent maps do not line up with much precision because the separate surveys may have been conducted in different times and by different survey groups. Without harmonising soil maps prior to the creation of soil attribute maps, artefacts such as boundary lines between different surveys become evident, which in the field are very unlikely to occur.

**Outcomes**

By the end of this exercise:

- You will be more familiar with some approaches for creating soil attribute maps from legacy soil maps.
- You will become aware of a number pre-processing approaches for up-dating, disaggregating, and harmonising legacy soil maps.
- You will have had some practical experience in a basic approach for creating a desktop soil class map from using ancillary data and cluster methods.
- You will have had some practical experiences in performing simple disaggregation techniques.

**Methods**

**Part 1. Deriving soil attribute maps from soil class maps**
This part of the exercise will be discussion orientated. We will look at some common examples for creating soil property maps from map unit/class maps. To do this, the popular approach is to use the physical and chemical data from the modal soil profiles of the soil classes/components. Where more than one class/component exists within a map unit, the common method is to derive the soil properties using a weighted average based on the areal proportions of each component in each map units. Other approaches for deriving soil property maps have this type of approach at the fundamental level.

**Part 2. Update existing soil maps**

In this part of the exercise we will discuss some examples of harmonising and/or updating soil maps using environmental covariates. Such approaches include those based on expert knowledge for example, SoLIM (Soil Land Inference Model) (Zhu and Band 1994).

A similar type approach which instead requires little expert knowledge, but rather is an empirical approach using environmental data layers is that of clustering. To illustrate, we will perform a clustering procedure (Kmeans) on environmental data layers into potential soil units. The data we will use covers an extent of the Lower Hunter Valley, near Cessnock, NSW. The existing soil units for this area are the DLWC soil landscapes. The idea is to use without any existing prior knowledge the environmental data layers to approximate the spatial extent of these DLWC soil units. We will use JMP for initial hard Kmeans clustering. The file required is Hunter Valley Covariates.jmp which is located in:

**\:\Sydney_DSM2011\Exercises\disaggregation**

We can also perform a similar clustering method, but this time with a fuzzy clustering approach whereby each point is given a membership grade to all clusters. This type of clustering approach could be considered better because it eliminates the presence of boundaries between classes and in the context of soil variability which exists continuously rather than discretely (predominantly) is logically a better pre-processing approach to ultimately make maps of soil properties. We will perform fuzzy clustering using the same data set using the FuzMe software. The file required here is HunterValleyCovariates_Fuzme.txt and is also located in the folder
described above. This example will be worked through ‘live’ on screen so that everyone becomes familiarised with the various software components.

Finally, other approaches for harmonising/updating soil maps will be discussed which include data mining and regression techniques. For these approaches we will discuss these in the context of work done by Bui and Moran (2003) for the Murray Darling Basin and by Grinand et al. (2007) in France.

**Part 3. Soil map disaggregation**

For this part of the exercise we will investigate some historical examples of soil map disaggregation techniques whereby we try to retrieve and dismantle some of the instilled soil landscape model information into individualised components and/or classes. Using the implicit information of the soil map one can use expert knowledge methods combined with ancillary environmental data to disaggregate mapping units. This is well described in Bui and Moran (2001) whom used the soil map unit descriptions in combination with environmental data and a clustering algorithm (Kmeans) to disaggregate soil mapping units. Other examples are discussed.

For some practical experience in soil map disaggregation (using the Kmeans) we will perform an empirical approach to disaggregation where, given the number of components within each map unit we generate the required number clusters using some ancillary environmental information example form the USA by performing. Of course the assumption with this method is that the distribution of the map unit components is a function of the given environmental data. Thus in this example we are working from the viewpoint that no instilled information is available other than the map units and their component distributions.

**Procedure**

In this part of the exercise we will use R for the implementation of Kmeans and then SAGA GIS to visualise the outputs. The data used for this example come from small section land with the extent of the Appanoose County, Iowa USA. We have clipped the SSURGO 1:25000 soil map sheet for this selected area. This can be observed in the figure below.
The labels within each of the polygons are the mapping unit labels. The information that we have about each of the map units is the number of individual components within each. Although in actuality there is a description of each component, we will be only concerned with the number at this stage.

- Navigate to the **:\Sydney\DSM2011\Exercises\disaggregation** folder in your directories. Here you will find a file called `Kmeans_disagg.r`. Double click on this and it will open up in the TinnR environment. Toggle `Ctrl+L` to display the line numbers of the script. Then click on the `R` icon in the toolbar to synchronise both the TinnR and R environments.

- **Lines 2 and 3** initialise the working directory and then call up the functions used for this script. Send these lines to R.

- **Lines 6 and 7** are instruction to read into R the soil data and ancillary information. The `MU_comp` object is a list of the map units in our area and the number of components each contains. The `dat` object is a data frame of the spatial information we have for this area. Observing the structure of this data frame, it is 30238 rows (in a spatial context these are pixels and the spatial resolution is 10m). Each column relates the the spatial position, map unit or terrain attribute. Send **Lines 6-8** to R.
Now send Line 11 to R. This essentially calls a looping function in which all the ancillary data is subset according to map unit. Then depending on the composition of each map unit, Kmeans clustering partitions the data into the same number of classes. Another column is added to the dat object where the component label is added for each row.

The output could be visualised quickly in R but instead better interaction with the outputs can be done using SAGA GIS. Send Line 14 to R. This saves an XYZ file for which will open up in SAGA.

To open this file in SAGA. First open up the software, then through the modules menu navigate to the Import Grid from XYZ tool.
When the new window pops up, you need to navigate the the soil_components.TXT file in the **:Sydney_DSM2011\Exercises\disaggregation folder from the file name field. Tick the has field names field, set the target cell size to 10m and the separator field the space. Click ok.

- Click on the resulting grid object that comes up in the data tab of the Workspace window. If you are familiar with the functionality of SAGA you can play around with the symbology aspects of the map to improve the visualisation of the map. Later on during the discussions of different approaches for soil map disaggregation we will look at another method of disaggregation (and harmonisation) where the same data set was used. You want to compare your results.
Given knowledge of the component properties of the map units it might be useful instead to perform fuzzy clustering instead of the hard Kmeans method. The example that we will work with now is a map unit from the above SSURGO soil map from Iowa. This map unit (406) has 3 components: - Kennebec (60%), Amana (30%) and Zook (10%). We will perform fuzzy clustering using the FuzMe software. The file required here is `map_unit_406.txt` and is also located in the folder described above. This example will be worked through ‘live’ on screen so that everyone becomes familiarised with the various software components. One clustering is complete it might be useful to visualise some of the outputs form this procedure in JMP. You will require the spatial coordinates for this however for which you will also need to open `map_unit_406_work.jmp` and then append to it the output from FuzMe.

In terms of other approaches for soil map disaggregation we will also discuss data mining techniques. We will briefly outline a method under development which simultaneously harmonises adjacent soil maps and then disaggregates the mapping units into component classes given as soil series. This work was conducted in the USA, and we will briefly outline the steps of this work that ultimately lead to the
creating of soil property map depicting both the lateral and vertical variation of soil properties organic carbon and pH.

References


END OF EXERCISE