Global Soil Salinity Map
GSS map

LESSON 3
Spatial modelling of soil indicators (properties) related to salt problems

GLOBAL SOIL PARTNERSHIP

Global Soil Salinity Map
GSS map

LESSON 3
Spatial modelling
of soil indicators (properties)
related to salt problems

Christian Thine Omuto
Ronald Vargas
Kostiantyn Viatkin
Yusuf Yigini

Contents

Figures

Tables

Summary

This Lesson is the third step of the capacity-building program, which is designed to build national capacities as well as harmonize procedures for developing information of salt-affected soils at the national and global levels. The overall goal of this Lesion is to support participants in developing spatial gridded maps of soil properties (indicators) related to salt-affected soils at the national level. At the end of the lesson, the participants are expected to have technical capacity in generating spatial information on soil indicators of salt-affected soils in their countries.

Summary outputs of gridded soil properties for establishing national information of salt-affected soils

Item	Description	Format
Gridded soil property maps	Topsoil (0-30 cm) EC, pH, and ESP	geoTiff raster files
	Subsoil (30-100 cm) EC, pH and ESP	geoTiff raster file
Gridded uncertainty maps	Topsoil (0-30 cm) EC, pH, and ESP	geoTiff raster files
	Subsoil (30-100 cm) EC, pH and ESP	geoTiff raster file
Textfile	Topsoil (0-30 cm) accuracy indices (ME, RMSE, r2, NSE)	Spreadsheet
	Subsoil (0-30 cm) accuracy indices (ME, RMSE, r2, NSE)	Spreadsheet

©FAO/Christian Omuto

1 Introduction

1.1 Overview

Many methods exist in the literature for mapping salt-affected soils. They include methods based on soil-type maps, remote sensing images, expert opinion, and digital soil mapping (DSM) of soil properties related to salt problems. The approach using DSM of soil indicators has the potential to quantify uncertainty and mapping accuracy besides developing spatial information (maps) of soil properties related to salt problems. This Lesson focuses on spatial modelling of soil indicator of salt problems in the soil. The Lesson also outlines steps for assessing accuracy and uncertainties associated with spatial modelling of soil indicators of salt-affected soils. It targets national experts with knowledge of and access to data on the indicators of salt-affected soils in their countries. Its outputs are expected to contribute to the development of national and global spatial information of salt-affected soils.

1.2 Objective

The overall objective of this Lesson is to spatially model soil indicators of salt problems using digital soil mapping approach.

1.3 Expected outcomes

By the end of this Lesson, the participants are expected to be able to:

  i.   Spatially model soil properties related to salt-affected soils using digital soil mapping approach
  ii. Assess soil mapping accuracy and uncertainty
  iii. Produce gridded maps of soil indicators (properties) related to salt problems in the soil

2 Requirements for assessing salt-affected soils

2.1 Data requirements

This Lesson uses test data, which was collected from the North State of Sudan (Figure 1). The data comprise

- Soil data (electrical conductivity (EC), pH and Exchangeable Sodium Percent (ESP))
- Spatial covariates such as mean annual rainfall amounts, land cover, geology, hydrogeology, MODIS remote sensing images, altitude (DEM)

The soil data is arranged as shown in Table 1.

Table 1: Organization of test case data

sample	Pits	Longitude	Latitude	Upper	Lower	Horizon	EC	pH	ESP
5	2	29.81	20.62	0	10	1.000	1.900	8.600	10.000
6	2	29.81	20.62	10	30	2.000	0.700	7.800	5.000
7	2	29.81	20.62	30	100	3.000			
8	3	31.57	17.15	0	35	1.000	0.900	7.600	5.000
9	3	31.57	17.15	35	60	2.000	0.400	7.800	2.000
10	3	31.57	17.15	60	100	3.000	0.400	7.900	2.000
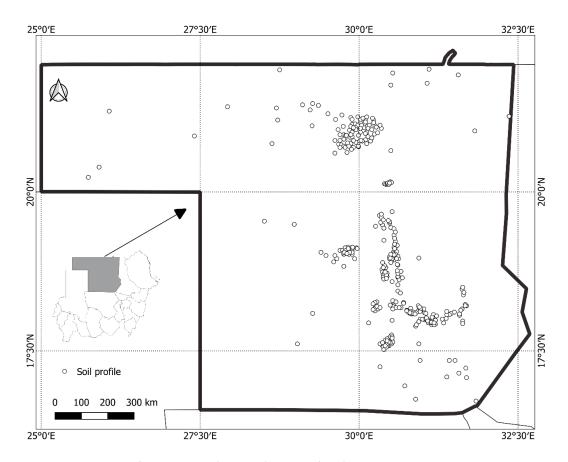
Figure 1: Location of test case soil in North State of Sudan

Other GIS files are raster files at a spatial resolution of 1 km and projected as WGS 84 (UTM 36N). Remote sensing images are corrected MODIS reflectance bands (Table 2).

Table 2: MODIS remote sensing images

Image	Spectral bands	
MODIS MOD009GA V6	Band 3	Blue: 0.459-0.479 μm
	Band 4	Green: 0.545-0.565 μm
	Band 1	Red: 0.62-0.67 μm
	Band 2	NIR: 0.841-0.876 μm
	Band 6	SWIR1: 1.628-1.652 μm
	Band 7	SWIR2: 2.105-2.13 μm

The test-case data (soil.RData, predictors.RData) have been stored (here) in R datafile format. Soil.RData is the calibration soil dataset while predictors.RData is a stack of GIS raster file.

2.2 Software requirements

The latest version of the software should have been installed (from Lesson 2)

i.   R (https://www.r-project.org/)
ii. QGIS (https://qgis.org/en/site/forusers/download.html)
iii. RStudio (https://rstudio.com/products/rstudio/download/#download)
iv. ILWIS (https://www.itc.nl/ilwis/download/ilwis33/)
v.   Spreadsheet software (Excel, Access) and document software (Word, Notepad)

R packages are also needed for spatial modelling with R: *soilassessment, sp, foreign, rgdal, car, carData, spacetime, gstat, automap, randomForest, e1071, caret, raster, soiltexture, GSIF, aqp, plyr, Hmisc, corrplot, factoextra, spup, purrr, lattice, ncf, ranger*. They should be downloaded and installed alongside R software.

3   Resources

The following resources are useful for implementing the activities during data collection:

- References
  - Technical guidelines and manual for mapping salt-affected soils (GSP-Secretariat@fao.org)
  - Country guidelines and specifications for global mapping of salt-affected soils

4   Activities

4.1   Loading data and R packages

#Step 1: Load the data and set working directory
Create a folder in C and call it Salinity (C:/Salinity) by right-click on New Folder in C using windows explorer (Figure 2). Unzip the downloaded zipped file (DSM_saltaffected.zip) in C/Salinity
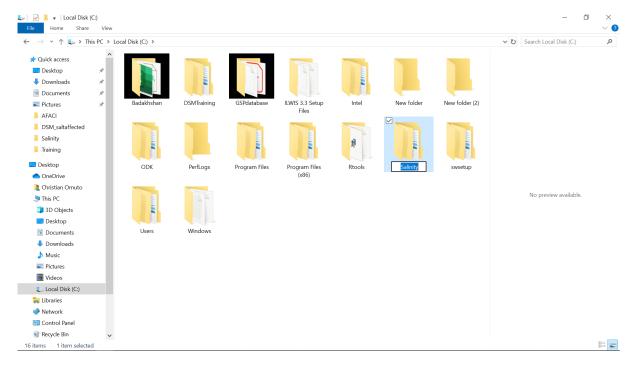
Figure 2: Creating working folder

In RStudio and at the top-right corner, click on project and scroll down to "Open Project". Navigate to C:/Salinity/DSM_saltaffected to locate "DSM_salaffected.Rproj" file. Choose it to load the project. In the bottom-right corner, choose *File* button (the first button in a set of File, Plots, Packages, Help, Viewer). This will reveal a set of files in the DSM_saltaffected.Rproj (Figure 3). Double click the files (one at a time) and accept the dialogue that follows (Digital_mapping_of_saltaffected_soils.R, predictors.RData, soil.RData, soilvalid.RData)
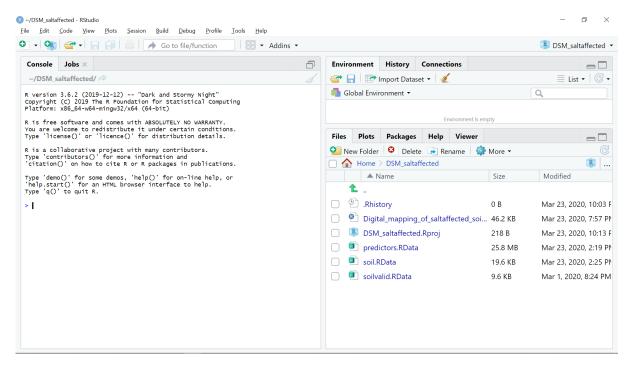
Figure 3: Loading files from R Project

At the top of the bottom-right panel, click the (gear) icon labelled *"More"*, and scroll to choose "*Set as Working Directory*" (Figure 4). This step sets the working directory of the test data.
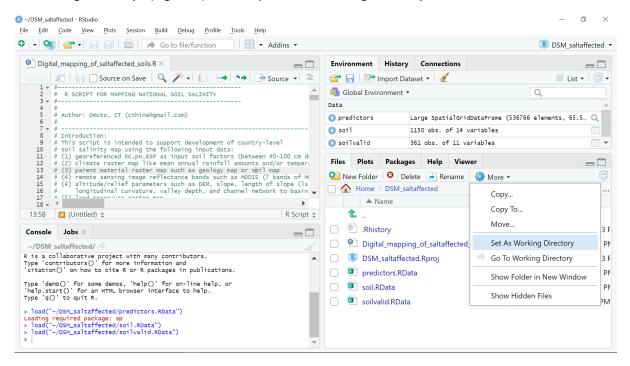
Figure 4: Setting working directory for test data
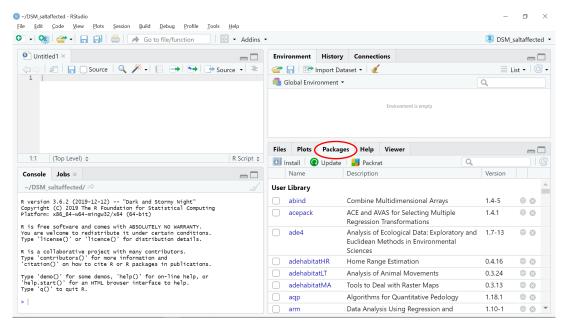
#Load R packages

R packages contain the functions for digitals soil mapping of soil properties. If the packages were not installed during Lesson 2, then they should be first installed before loading the libraries.
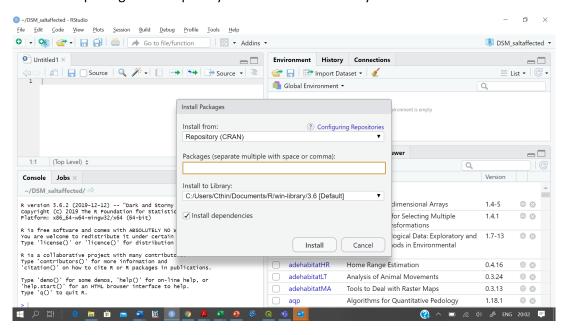
```
> install.packages(c("raster", "sp", "rgdal", "car", "carData", "dplyr", "spacetime
", "gstat", "automap", "randomForest", "fitdistrplus", "e1071", "caret", "soilasses
sment", "soiltexture", "GSIF", "aqp", "plyr", "Hmisc", "corrplot", "factoextra", "s
pup", "purrr", "lattice", "ncf", "npsurv", "lsei", "qrnn", "nnet", "mda", "RColorBr
ewer", "vcd", "readxls","maptools","neuralnet","psych","ggrepel", "plotly"))
```

In case there is error in installation,

1. Check if internet connectivity is adequate
2. If internet connectivity is OK, then note the packages which are not installing and install them manually
3. click packages in the lower bottom panel

4. Click *Install* icon below the *Packages* button in (3) above. A window will pop-up for installing the package. At this point your internet connectivity should be active

5. Type the name of the package in the space below *Packages (separate multiple with space or comma): (NB: you can also c*opy and paste each bullet below into the same space)

<ol type="a">
raster, sp, rgdal, car, carData, dplyr, spacetime, gstat, automap, randomForest, fitdistrplus, e1071,
caret, soilassessment, soiltexture, GSIF, aqp, plyr, Hmisc, corrplot, factoextra, spup, purrr, lattice
ncf, npsurv, lsei, qrnn, nnet, mda, RColorBrewer, vcd, readxls, maptools, neuralnet, psych, ggrepel, plotly

The libraries should be loaded after installing the packaged

```
>library(sp);library(foreign);library(rgdal);library(car);library(carData);library(
maptools)
>library(spacetime);library(gstat);library(automap);library(randomForest);library(f
itdistrplus);
>library(e1071);library(caret);library(raster);library(soilassessment);library(soil
texture);
>library(GSIF);library(aqp);library(plyr);library(Hmisc);library(corrplot);library(
factoextra)
>library(spup);library(purrr);library(lattice);library(ncf);library(npsurv);library
(lsei);
>library(nnet);library(class);library(mda);library(RColorBrewer);library(vcd);libra
ry(grid);
>library(neuralnet);library(readxl);library(psych);library(qrnn);library(dplyr)
```

4.2 Check and harmonize statistical distribution of GIS layers

Before checking and harmonizing statistical distribution of GIS layers, it is important to check and remove pixels with data (NA pixels).

#Check and remove NA

```
> summary(predictors)
Object of class SpatialGridDataFrame
Coordinates:
        min        max
x -356126.8  465873.2
y 1825343.5 2478343.5
Is projected: TRUE
proj4string :
[+proj=utm +zone=36 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0]
Grid attributes:
  cellcentre.offset cellsize cells.dim
x         -355626.8     1000       822
y         1825843.5     1000       653
Data attributes:
…..

    lcover          geology         pgeology          rain            swir1
Min.   :  2.0  Min.   : 1.00  Min.   : 0.9977  Min.   : 0.1938  Min.   :0.01449
1st Qu.:178.0  1st Qu.:31.00  1st Qu.: 3.0000  1st Qu.: 2.0000  1st Qu.:0.58141
Median :178.0  Median :32.00  Median : 3.0000  Median : 4.0000  Median :0.65651
Mean   :177.4  Mean   :47.55  Mean   : 3.8039  Mean   : 7.2899  Mean   :0.63076
3rd Qu.:178.0  3rd Qu.:66.00  3rd Qu.: 3.0000  3rd Qu.: 7.8400  3rd Qu.:0.70472
Max.   :188.0  Max.   :88.00  Max.   :10.0000  Max.   :70.5665  Max.   :0.93891

…..
```

NAs will be shown in the layers where they occur. They should be first investigated where/why they occur. If they are predominantly outside the study area and occurred dude to GIS raster clipping, then they can be removed, for example, by replacing them with the mean of the data.

```
> predictors$slope=ifelse(is.na(predictors$slope),mean(!is.na(predictors$slope)),pr
edictors$slope)
```

6

#Derive the remote sensing indices

Derive the remote sensing indices of salt problems and attach them in the *predicters* stack of GIS layers. The indices are derived using the function *imageIndices* in the *soilassessment* library (Omuto, 2020[1]).

```
> predictors$SI1=imageIndices(predictors$BBlue,predictors$BGreen,predictors$BRed,pr
edictors$BIRed,predictors$swir1,predictors$swir2,"SI1");summary(predictors$SI1)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.04471 0.31981 0.35173 0.34386 0.37887 0.54641

> predictors$SI2=imageIndices(predictors$BBlue,predictors$BGreen,predictors$BRed,pr
edictors$BIRed,predictors$swir1,predictors$swir2,"SI2");summary(predictors$SI2)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0389  0.2297  0.2540  0.2490  0.2749  0.4108
.......

> predictors$BI=imageIndices(predictors$BBlue,predictors$BGreen,predictors$BRed,pre
dictors$BIRed,predictors$swir1,predictors$swir2,"BI");summary(predictors$BI)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.07189 0.73497 0.83005 0.80074 0.89241 1.22795
```

Any NAs arising from the calculation of the image indices should be remove where necessary.

#Check for skewness using empirical histogram distribution

```
> hist(predictors@data[,27:29]) # Figure 5.2
> summary(predictors$SI6)
#    Min.  1st Qu.   Median     Mean  3rd Qu.    Max.
#  0.003647 0.940943 1.129692 1.068303 1.232106 1.663694

> predictors$BI=sqrt(predictors$BI)
> hist(predictors$BI)
```


Figure 5: Empirical statistical distribution of image indices

For the moment, square-root or log transformation can be tested for data normalization.

```
> hist(predictors@data[,"rain"])
> summary(predictors$rain)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1938  2.0000  4.0000  7.2899  7.8400 70.5665
> predictors$rain=log(predictors$rain)
```

---

[1]Omuto, CT. 2020. soilassessment: Assessment Models for Agriculture Soil Conditions and Crop Suitability. https://cran.r-project.org/web/packages/soilassessment/index.html

After normalizing the image indices, they are selected and converted into data-frame to enable determination of correlation and principal component analysis. Afterwards, the selected PCs are converted back to the raster stack.

 # Extract the image layers

```
> predicters=predicters@data[,c("SI1","SI2","SI3","SI4","SI5","SI6","SAVI","VSSI","NDSI","NDVI","SR","CRSI", "BI")]
> soil.cor=cor(predicters)
> corrplot(soil.cor,method="number",number.cex = 0.8) # Figure 6a
> pca<-prcomp(predicters[], scale=TRUE)
> fviz_eig(pca) # Figure 6b
```

The correlation plot (Figure 6a) shows the correlation between image indices. For example, SI1 and SI2 have Pearson correlation index equal to 86%. PCA examines these correlations and determines the principal axes where data are highly correlated. These axes are also known as principal component (or dimensions in Figure 6b). Figure 6 is important in guiding the choice of PCs to represent the entire (13) layers of image indices.
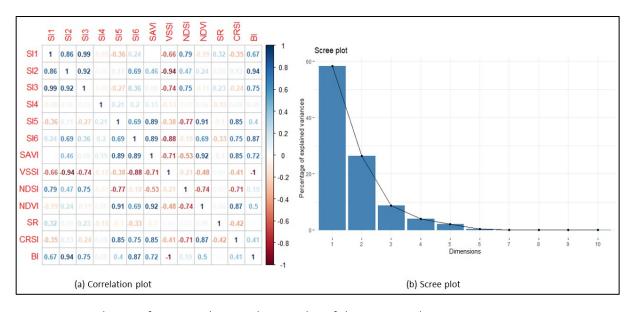
Figure 6: Correlation of image indices and scree plot of their principal component

In Figure 6b, cumulative sum of the first 5 PCs (Dimensions) add to up to more than 95% explained variation in the overall 13 layers of the image indices. Hence, the first 4 PCs can adequately represent the 13 image indices. This approach can be used on any set of image indices to select the appropriate number of PCs to reduce the data bulk.

 # Return the selected PCs to the raster stack to complete the harmonization process

```
> Pred.pcs<-predict(pca,predicters[])
> predictors@data$PCA1=Pred.pcs[,1]
> predictors@data$PCA2=Pred.pcs[,2]
> predictors@data$PCA3=Pred.pcs[,3]
> predictors@data$PCA4=Pred.pcs[,4]
```

4.3 Harmonization of input soil data
 # Harmonize input indicator measurements to those for saturated soil paste extract

Many methods can be used to determine EC. They include (1) the use of saturated soil paste extract, (2) using other extracts, (3) using pedotransfer models from other soil properties, or (4) electromagnetic induction. Harmonization seeks to standardize methods 2 to 4 to equivalent values in method 1, since popular classification schemes use values obtained by method 1. The test case data was determined using saturated soil paste extract. Hence, it will not require EC harmonization. Omuto et al. (2020)[2] outlines steps for harmonizing EC for the cases 2 to 4.

Soil depth harmonization aims at developing soil information for uniform depth throughout the soil data. This harmonization is achieved by using the depth-integrating spline approach (Bishop et al., 1999[3]). The tool for implementing the approach is contained in the *GSIF* package (Hengl, 2019)[4].

```
> lon=soil1$Longitude
> lat=soil1$Latitude
> id=soil1$Pits
> top=soil1$Upper
> bottom=soil1$Lower
> horizon=soil1$Horizon
> ECdp=soil1$EC
> prof1=join(data.frame(id,top,bottom, ECdp, horizon),data.frame(id,lon,lat),type="inner")
Joining by: id
> depths(prof1)=id~top+bottom
Warning message:
converting IDs from factor to character
> site(prof1)=~lon+lat
> coordinates(prof1) = ~lon+lat
> proj4string(prof1)=CRS("+proj=longlat +datum=WGS84 +no_defs")
> depth.s = mpspline(prof1, var.name= "ECdp", lam=0.8,d = t(c(0,30,100,150)))
Fitting mass preserving splines per profile...

   |================================================================| 100%
> plot(prof1, color= "ECdp", name="horizon",color.palette = rev(brewer.pal(8, 'Accent')),par=c(cex.lab=2.0)) #Figure 7
```


Figure 7: Example depth harmonization for EC$_{se}$

---

[2]Omuto CT, Vargas RR, El Mobarak, AM, Mohamed N, Viatkin K, Yigini Y. (Eds). 2020. Global mapping of salt-affected soils: A technical guideline and cookbook. Rome

[3]Bishop, T.F.A, McBratney, A.B., Laslett, G.M., 1999. Modelling soil attribute depth functions with equal-area quadratic smoothing splines. Geoderma, 91(1-2), 27-45

[4] Hengl T. 2019. GSIF: Global Soil Information Facilities. https://cran.r-project.org/web/packages/GSIF/index.html

 # Extract the depth-harmonized soil data and re-project

```
> soilhrmdepths=data.frame(depth.s$idcol, depth.s$var.std, check.names = TRUE)
> soil2=merge(soil1,soilhrmdepths,by=intersect(names(soil1),names(soilhrmdepths)),by.x="Pits",by.y="depth.s.idcol",all=TRUE)
> coordinates(soil2)=~Longitude+Latitude
> proj4string(soil2)=CRS("+proj=longlat +datum=WGS84")#Attach CRS to the data
```

#Harmonize CRS and ensure use of the correct +proj and +zone for the study area

```
> soil1=spTransform(soil2,CRS("+proj=utm +zone=36 +ellps=WGS84 +units=m +no_defs"))
> soil1=soil2
> hist(soil1$EC)
> soil1=subset(soil1,!is.na(soil1$EC))
```

#Harmonization of statistical distribution

This harmonization is done to transform the frequency distribution to normal distribution. Frequency transformation to normal distribution is optional for spatial modelling algorithms. If it's chosen, then the empirical distribution is first established through histogram analysis and transformation implemented if the distribution is found to be skewed. *hist* function is used to extract and plot the histogram. Box-Cox (1964) transformation is preferred. The following scripts illustrate the steps for transforming statistical distribution. Summary distribution is first obtained to establish if there are zeros, *Nas*, or negative values. It is desirable to remove them before implementing Box-Cox transformation.

```
> summary(soil1$X0.30.cm)
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.0007   0.6291   1.8709   6.6812   5.3121 154.2463
```

```
> soil1$dummy=(soil1$EC)# (you may) add "+0.001" if minimum X0.30.cm is zero
> hist(soil1$dummy, main="Frequency distribution (before transformation)", xlab="Harmonized EC (dS/m)")
> soil1$Tran=(soil1$dummy^(as.numeric(car::powerTransform(soil1$dummy, family ="bcPower")["lambda"]))-1)/(as.numeric(car::powerTransform(soil1$dummy, family ="bcPower")["lambda"]))
```

4.4   Spatial modelling of indicators

Spatial modelling of indicators of salt-affected soils is based on the digital soil mapping (DSM) concept. In this concept, a relationship is built between the soil indicators of salt problems and spatial predictors (GIS layers of drivers and indicators of salt problems and soil forming factors). This approach enables quantification of:

1. Spatial information of indicators of salt-affected soils (EC, pH, ESP) and different soil depths
2. Mapping uncertainties and accuracy
3. Spatial information of classes and intensity of salt problems

Popular models often used to represent *f* are linear, random-forest, support-vector machine, mixed-effects, regression kriging, etc. The *soilassessment* package provides *regmodelSuit* function for guiding the choice of the appropriate model for mapping soil variables. It tests different models and returns the top nine models using RMSE, ME, NSE and r^2. Lowest root mean-square error (RMSE), highest r^2, lowest mean error (ME), highest Nash-Sutcliff coefficient of efficiency (NSE) are then used as the guiding criteria for choosing the suitable model. An initial step for spatial modelling is to build the model in a calibration dataset and then testing the model using an independent dataset. This calls for the establishment of calibration and validation datasets. These datasets should have well aligned soil properties (indicators) and spatial predictors at each georeferenced sampling point. Pixel value

extraction of GIS layers (predictors) using point data (soil1) is a suitable method for developing either the calibration or validation datasets.

 # First check for similarity in coordinate reference system – crs and then extract the predictors

```
> crs(predictors); crs(soil1)
CRS arguments:
 +proj=utm +zone=36 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
CRS arguments:
 +proj=utm +zone=36 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
```

It's important to ensure that the CRS for predictors and soil database are the same before starting pixel extraction

#Then extract the pixel values for all predictors into the soildata dataframe

```
> {predictors.ov=over(soil1, predictors)
+    soil1$dem=predictors.ov$dem
+    soil1$slope=predictors.ov$slope
+    soil1$cnbl=predictors.ov$cnbl
+    soil1$ls=predictors.ov$ls
+    soil1$valley=predictors.ov$valley
+    soil1$loncurve=predictors.ov$loncurve
+    soil1$lcover=predictors.ov$lcover
+    soil1$rain=predictors.ov$rain
+    soil1$pgeology=predictors.ov$pgeology
+    soil1$geology=predictors.ov$geology
+    soil1$PCA1=predictors.ov$PCA1
+    soil1$PCA2=predictors.ov$PCA2
+    soil1$PCA3=predictors.ov$PCA3
+    soil1$PCA4=predictors.ov$PCA4
+ }
```

#Step 2-2: Establish suitable DSM model

```
> soil1=subset(soil1,!is.na(soil1$dem))
> soil11a=soil1@data[,c("Tran","dem","slope","ls","cnbl","loncurve","valley","rain"
,"lcover","pgeology","geology","PCA1","PCA2","PCA3","PCA4")]
> regmodelSuit(soil11a,Tran,dem,geology,pgeology,slope,rain,loncurve,cnbl,valley,lc
over,ls,PCA1,PCA2,PCA3, PCA4)
  |=================================================================| 100%
                  ME        RMSE       R2         NSE
Linear          1.37034834 1.8129133 0.1320264 -4.45424486
RandomForest    0.24614749 0.4291176 0.9623631  0.99707891
SVM             1.34745209 1.8212570 0.1357457 -4.44014767
BayesianGLM     1.36669809 1.8051662 0.1399516 -4.55079779
BaggedCART      0.88676091 1.1705841 0.7018074  0.44302759
Cubist          0.07851255 0.2744213 0.9753726  1.00000000
CART            1.40147986 1.8274332 0.1320797 -4.56273851
Ranger          0.26852953 0.4205875 0.9655103  0.99702690
QuantRandForest 0.04923343 0.2855093 0.9761419  1.00000000
QuantNeuralNT   1.16162791 1.7075411 0.2582558  0.07778314
```

The above results depict the *quantum regression random forest* and *cubist* models as suitable for modelling the 0-30cm EC$_{se}$ using the given spatial predictors in the case-study test data.

Statistical model building and testing strategies recommend independent datasets for model building and for model testing. These datasets should ideally be sampled with focus for model building and testing. In the absence of independently sampled dataset for either model building (calibration) or testing (validation), data-splitting strategy is often used. Data-splitting strategy randomly (or stratified randomly) splits the data into two parts. One part is held as calibration and the other as validation. The validation dataset is used for accuracy assessment. The indices for reporting modelling accuracy include RMSE, ME, r^2, NSE, and a graphical plot of the modelled versus harmonized values. Data-splitting may be arbitrarily chosen according to or depending on the data-size.

```
> {soil4=as.data.frame(soil1)
+    bound <- floor((nrow(soil4)/4)*3)
+    soil3 <- soil4[sample(nrow(soil4)), ]
+    df.traina <- soil3[1:bound, ]
+    df.testa <- soil3[(bound+1):nrow(soil3), ]}

> rf.ec=train(Tran~(slope+rain+loncurve+ls+cnbl+valley+lcover+dem+PCA1+PCA2+PCA3+PC
A4+PCA5),  data = df.traina,  method = "qrf", trControl=trainControl( method = "cv"
,number=5,returnResamp = "all",savePredictions = TRUE, search = "random",verboseIte
r = FALSE))

# Show the prediction interval
> df.testa$Strain=predict(rf.ec,newdata=df.testa)
> hist(df.testa$Strain,xlab="Box-Cox Transformed ECse (0-30cm)", main=NULL)
> abline(v = quantile(df.testa$Strain, probs = c(0.05, 0.95)),lty = 5, col = "red")
```


Figure 8: Graphical plot of frequency distribution with prediction limits at 95% confidence interval

Prediction limits on the Box-Cox transformed values at 95% confidence interval are given in Figure 8, which shows the interval around the mean of 0.77 as [-2, 4.1].

#Accuracy assessment

```
> cor(df.testa$Strain,df.testa$dummy)^2
[1] 0.9950319

> {plot(df.testa$Strain~df.testa$dummy, xlab="Measured ECse",ylab="Modelled ECse",
main="Accuracy assessment on hold-out samples")
+    abline(a=0,b=1,lty=20, col="blue")} # Figure 9
```
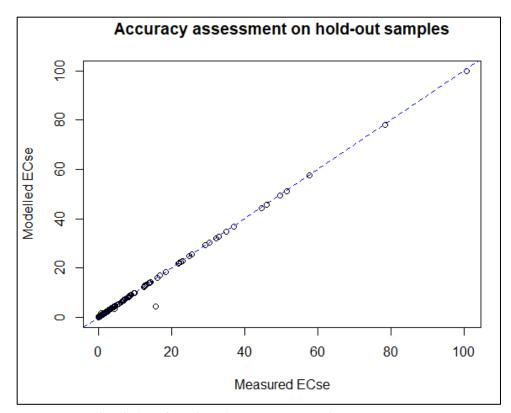
Figure 9: Graphical plot of predicted versus measured EC

```
> Bias=mean(df.testa$Strain-df.testa$dummy,na.rm=TRUE)
> RMSE=sqrt(sum(df.testa$Strain-df.testa$dummy,na.rm=TRUE)^2/length((df.testa$Strai
n-df.testa$dummy)))
> Rsquared=cor(df.testa$Strain,df.testa$dummy)^2
> NSE=1-sum(df.testa$Strain-df.testa$dummy,na.rm=TRUE)^2/sum((df.testa$Strain-mean(
df.testa$dummy,na.rm=TRUE))^2,na.rm=TRUE)
> statia=data.frame(Bias,RMSE,Rsquared,NSE);View(statia)
> write.csv(statia,file = "EC0_30_validmodel_stats.csv")
> statia
         Bias      RMSE  Rsquared       NSE
1 -0.1019564 1.751158 0.9950319 0.982046
```

#Use the developed model to predict the map of EC

```
> lmbda1=(as.numeric(powerTransform(soil1$dummy, family ="bcPower")["lambda"]))
> predictors$ECte=predicta(rf.ec,predictors)
> coordinates(df.testa)=~Longitude+Latitude
> proj4string(df.testa)=CRS("+proj=utm +zone=36 +datum=WGS84 +units=m +no_defs +ell
ps=WGS84 +towgs84=0,0,0") # Make sure to use correct CRS
> predicters.ov1=over(df.testa, predictors)
> df.testa$Predre=predicters.ov1$ECse
> cor(df.testa$dummy,df.testa$Predre)^2
[1] 0.9978655
```

#Compare the spatial prediction and validation dataset

```
> featureRep(predictors["ECse"],df.testa) #Figure 5.10

        Two-sample Kolmogorov-Smirnov test

data:  dist.histbb$left and dist.histbb$right
D = 0.52174, p-value = 0.003819
alternative hypothesis: two-sided

> summary(predictors$ECse);summary(df.testa$dummy)
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.00007   0.48810   1.17487   1.51685   1.61781 112.74435
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.00048   0.59755   1.71126   6.60388   5.05220 113.50941
```
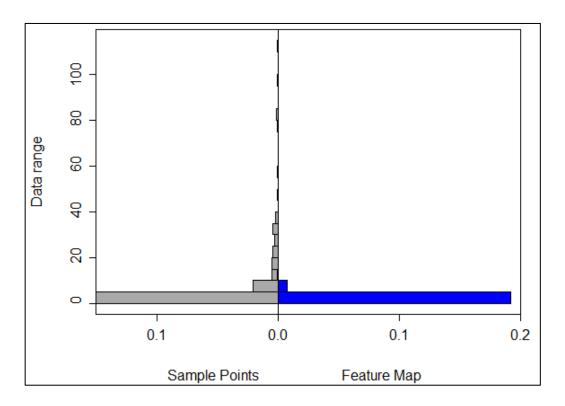
13

Figure 10: Representativeness of validation (sample points) EC ranges in prediction map (feature map)

The feature representation shows how well the range of measured EC (validation EC) are contained in the prediction map. In the case-study sample, high EC (>40 dS/m) are poorly captured in the prediction map. The x-axis shows the frequency (probability density) of occurrence of data (EC) values in y-axis. Poor representation of the high (EC > 40) implies model uncertainty for high EC values. This will be further investigated when uncertainties are produced.

#Export the output

writeGDAL(predictors["ECse"], drivername = "GTiff", "Top0_30ECse.tif")

#Uncertainty assessment
```
> soil6a=soil1[,c("Tran")]
> predictors6a=predictors[c("dem","slope","cnbl","lcover","loncurve","rain","pgeolo
gy","geology","ls","valley","PCA1","PCA2","PCA3","PCA4","PCA5")]

> pred_uncerta=predUncertain(soil6a,predictors6a,3,95,"qrandomforest")
  |==========================================================| 100%

> spplot(pred_uncerta, "pred_width", scales = list(draw = TRUE),col.regions=heat.co
lors(20,rev = TRUE)) + spplot(df.testa,"dummy",pch=3,cex=0.4) #Figure 11
```

#Step 2-7: Exporting the uncertainty maps

```
> EC0_30_uncertain=(pred_uncerta$pred_width*lmbda1+1)^(1/lmbda1)
> writeRaster(EC0_30_uncertain, filename="EC0_30_uncertain.tif",format="GTiff")
```
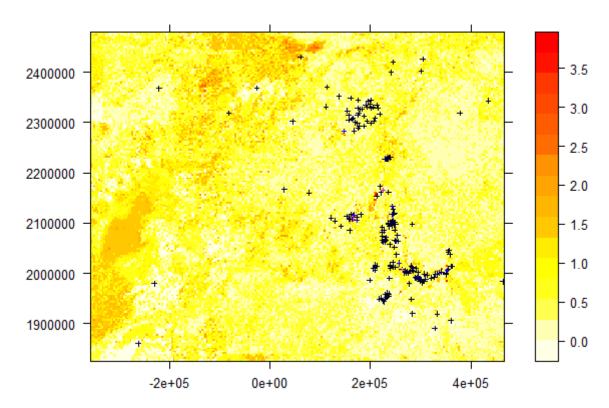
Figure 11: Spatial prediction width at 95% confidence interval and overlay of validation points

The above steps for spatial modelling of EC should be repeated for pH, ESP for 30-100 cm soil depths.

5 Outputs

Each participant is expected to produce the following at the end of this lesson:

1. GeoTiff raster maps of soil indicators of salt-affected soils (EC, pH, anf ESP) for 0-30 cm and 30-100 cm (all together 6 raster maps)
2. GeoTiff raster maps of uncertainty assessment for each soil property (EC, pH, and ESP) for 0-30 cm and 30-100 cm (all together 6 raster maps)
3. Text file of accuracy assessment for each soil property (EC, pH, and ESP) for 0-30 cm and 30-100 cm (all together 6 raster maps)

CA9202EN/1/07.20