7

Soil Organic Carbon (WISE_SOC)

7.1 Introduction

ISRIC WISE is an international soil profile data set; a selection of globally distributed soil profiles, prepared by the International Soil Reference and Information Centre (ISRIC) located in Wageningen (Batjes, 2008, 2009). The most recent version (3.1) of the soil profile database contains 10,253 profiles[1]. The database consists of several tables, the most important are: `WISE3_SITE` (information about the soil profile site) and `WISE3_HORIZON` (laboratory data per horizon). This chapter demonstrates how to estimate the global **Soil Organic Carbon** (SOC) stock using regression-kriging and a large repository of publicly accessible global environmental maps (about 10 km resolution) described in section 4.1. The results contribute to the GlobalSoilMap.net initiative that aims at producing high resolution images of key soil properties and functions (Sanchez et al., 2009).

The maps presented in this exercise were created for demonstration purposes only. The true accuracy/-consistency of these maps has not been evaluated and is heavily controlled by the representativeness of the sampling pattern and accuracy of individual measurements (refer to the ISRIC WISE general disclaimer[2]). Positional accuracy of profiles in WISE varies depending on the source materials from which they were derived — this may range from the nearest second Lat/Lon up to few meters. Most of the available legacy data considered in WISE date from the pre-GPS era. In addition, the list of predictors we use in this exercise could be much more extensive; many of the maps are also available at finer resolutions (∼1 km). Compare also the maps produced in this chapter with the global soil property maps distributed by ISRIC[3], and/or the Global Biomass Carbon Map produced by the CDIAC[4] (Ruesch and Gibbs, 2008b). Note that this is also a relatively large data set and computations can become time-consuming. It is not recommended to run this exercise using a PC without at least 2 GB of RAM and at least 1 GB of hard disk memory.

7.2 Loading the data

To run this script, you first need to register and obtain the MS Access file from the ISRIC website[5]. Before you start using the ISRIC WISE database, please also read about its limitations in Batjes (2008). Next, load the necessary packages:

```
> library(RODBC)
> library(gstat)
> library(rgdal)
> library(RSAGA)
> library(spatstat)
```

[1] Not all profiles are complete.
[2] http://www.isric.org/isric/webdocs/Docs/ISRIC_Report_2008_02.pdf
[3] http://www.isric.org/UK/About+Soils/Soil+data/Geographic+data/Global/WISE5by5minutes.htm
[4] http://cdiac.ornl.gov/epubs/ndp/global_carbon/carbon_documentation.html
[5] http://www.isric.org/isric/CheckRegistration.aspx?dataset=9

For more info on how to set-up SAGA GIS and run the commands from R see section 3.1.2.

7.2.1 Download of the world maps

Next, download and unzip all relevant predictors from the web-repository[6]:

```
# location of maps:
> URL <- "http://spatial-analyst.net/worldmaps/"
# list of maps:
>  map.list <- c("biocl01", "biocl02", "biocl04", "biocl05", "biocl06", "biocl12",
+     "biocl15", "countries", "dcoast", "globedem", "landcov", "landmask", "gcarb",
+     "nlights", "pcndvi1", "pcndvi2", "pcndvi3", "pcpopd1", "himpact", "glwd31",
+     "wildness", "hwsd", "quakein", "iflworld", "treecov")
# download the zipped maps one by one:
> for(i in 1:length(map.list)) {
>   download.file(paste(URL, map.list[i], ".zip", sep=""),
+        destfile=paste(getwd(), "/", map.list[i], ".zip", sep=""))
>   unzip(paste(getwd(), "/", map.list[i], ".zip", sep=""))
>   unlink(paste(map.list[i], ".zip", sep=""))
# Delete temporary file:
>   unlink(paste(map.list[i], ".zip", sep=""))
> }
```

```
trying URL 'http://spatial-analyst.net/worldmaps/biocl01.zip'
Content type 'application/zip' length 1362739 bytes (1.3 Mb)
opened URL
downloaded 1.3 Mb
...
```

where `biocl1-15` are long-term bioclimatic variables; `dcoast` is the distance from coastline; `globedem` is the ETOPO1 Global Relief Model; `landcov` is the Global Land Cover map of the world; `landmask` is the land mask; `gcarb` is the carbon (biomass) density in tones of C/ha; `nlights` is the long-term annual image of lights at night; `pcndvi1/2` are first and second principal component derived from 20 years of AVHRR NDVI monthly images; `pcpopd1` is PC1 of the Gridded Population of the World, version 3 (GPWv3), `himpact` is the world map of human impacts-free areas estimated by the GLOBIO initiative of the United Nations Environment Programme; `glwd31` is the indicator map showing location of wetlands based on the Global Lakes and Wetlands Database (GWLD3.1) database; `wildness` is a map of the World wilderness areas; `hwsd` is soil class map based on the FAO Harmonized World Soil Database v 1.1 (37 classes); `quakein` is the Earthquake intensity (magnitude) based on the NOAA's Significant Earthquake Database; `iflworld` is the world map of intact forest landscapes; `treecov` is the Vegetation percent tree cover (see also §4.1). If the download was successful, you will notice that the ArcInfo ASCII grids are now available in your working directory. A detailed description of each layer is available via the raster description (`*.rdc`) file. See p.159 for an example.

We can load some maps, that we will need later on, into R using rgdal:

```
> worldmaps <- readGDAL("landmask.asc")
```

```
landmask.asc has GDAL driver AAIGrid
and has 1300 rows and 3600 columns
```

```
> names(worldmaps) <- "landmask"
> worldmaps$landcov <- as.factor(readGDAL("landcov.asc")$band1)
> worldmaps$glwd31 <- as.factor(readGDAL("glwd31.asc")$band1)
> worldmaps$hwsd <- as.factor(readGDAL("hwsd.asc")$band1)
> proj4string(worldmaps) <- CRS("+proj=longlat +ellps=WGS84")
```

[6] http://spatial-analyst.net/worldmaps/

7.2.2 Reading the ISRIC WISE into R

If you have obtained the ISRIC-WISE_ver3.mdb file from ISRIC, you can connect to it by using the RODBC[7] package:

```
> cGSPD <- odbcConnectAccess("ISRIC-WISE_ver3.mdb")
# Look at available tables:
> sqlTables(cGSPD)$TABLE_NAME
```

```
 [1] "MSysAccessObjects"
 [2] "MSysAccessXML"
 [3] "MSysACEs"
 [4] "MSysObjects"
 [5] "MSysQueries"
 [6] "MSysRelationships"
 [7] "WISE3__ReadMeFirst"
 [8] "WISE3_coding_conventions"
 [9] "WISE3_HORIZON"
[10] "WISE3_LABcodes_Description"
[11] "WISE3_LABname"
[12] "WISE3_LABname_codes"
[13] "WISE3_SITE"
[14] "WISE3_SOURCE"
```

Now that we have connected to the database, we query it to obtain values from the tables like with any other SQL database. We need to obtain the following five variables:

- ORGC = Organic carbon content in *promille* (or g C kg^{-1});
- TOPDEP, BOTDEP = Thickness of soil horizons in cm;
- LON, LAT = Point coordinates;

We first fetch measured values for organic content and the depths of each horizon from WISE3_HORIZON table:

```
> GSPD.HOR <- sqlQuery(cGSPD, query="SELECT WISE3_ID, HONU, ORGC, TOPDEP,
+       BOTDEP FROM WISE3_HORIZON")
> str(GSPD.HOR)
```

```
'data.frame':   47833 obs. of  5 variables:
 $ WISE3_ID: Factor w/ 10253 levels "AF0001","AF0002",..: 1 1 1 2 2 2 2 3 3 3 ...
 $ HONU    : int  1 2 3 1 2 3 4 1 2 3 ...
 $ ORGC    : num  7.6 2.3 0.9 12.8 6 3.9 2.7 5.9 2.4 NA ...
 $ TOPDEP  : int  0 15 60 0 20 60 110 0 20 50 ...
 $ BOTDEP  : int  15 60 150 20 60 110 170 20 50 110 ...
```

```
# Horizon thickness:
> GSPD.HOR$HOTH <- GSPD.HOR$BOTDEP-GSPD.HOR$TOPDEP
# unique ID:
> GSPD.HOR$ID <- as.factor(paste(as.character(GSPD.HOR$WISE3_ID),
+        GSPD.HOR$HONU, sep="_"))
```

where HONU is the horizon number (from the soil surface) and TOPDEP and BOTDEP are the upper and lower horizon depths. This shows that there are over 45 thousand measurements of the four variables of interest. The number of soil profiles is in fact much smaller — as you can see from the WISE3_ID column (unique profile ID), there are 10,253 profiles in total.

We know from literature that total soil organic carbon (SOC) depends on bulk density of soil and coarse fragments (Batjes, 1996; Batjes et al., 2007). There is certainly a difference in how ORGC relates to SOC in

[7] http://cran.r-project.org/web/packages/RODBC/

volcanic soils, wetland soils, organic soils and well drained mineral soils. To correctly estimate total Soil Organic Carbon in kg C m^{-2}, we can use the following formula[8]:

$$\texttt{SOC}\ [\text{kg m}^{-2}] = \frac{\texttt{ORGC}}{1000}\ [\text{kg kg}^{-1}] \cdot \frac{\texttt{HOTH}}{100}\ [\text{m}] \cdot \texttt{BULKDENS} \cdot 1000\ [\text{kg m}^{-3}] \cdot \frac{100 - \texttt{GRAVEL}\ [\%]}{100} \qquad (7.2.1)$$

where BULKDENS is the soil bulk density[9] in g cm^{-3} and GRAVEL is the gravel content in profile expressed in %.

Because we are interested in total organic carbon content for soil profile, we will first estimate SOC values for all horizons, then aggregate these values per whole profile. Alternatively, one could try to predict organic carbon for various depths separately, then aggregate number of maps. Spatial analysis of soil layers makes sense because one can observe both shallow soils with high and low SOC content and vice versa, and these can both be formed under different environmental conditions. For the purpose of this exercise, we will focus only on the aggregate value i.e. on the total estimated soil organic carbon per profile location.

We can load an additional table[10] with Bulk density / gravel content estimated at fixed depth intervals (0–20, 20–40, 40–60, 60–80, 80–100, 100–150, 150–200 cm):

```
> load(url("http://spatial-analyst.net/book/system/files/GSPD_BDG.RData"))
> str(GSPD.BDG)

  'data.frame':   47111 obs. of  4 variables:
   $ WISE3_ID: Factor w/ 8189 levels "AF0001","AF0002",..: 1 1 1 1 1 1 2 2 2 2 ...
   $ BULKDENS: num  1.55 1.58 1.58 1.6 1.55 ...
   $ GRAVEL  : int  16 5 6 5 4 3 2 4 4 2 ...
   $ DEPTH   : num  10 30 50 70 90 125 10 30 50 70 ...
```

where BULKDENS is expressed in g cm^{-3}, GRAVEL is expressed in %, and DEPTH in cm. We first need to re-estimate the BULKDENS and GRAVEL at original depths for which we have ORGC measurements. We can do this by using e.g. linear interpolation:

```
# re-estimate values of BULKDENS and GRAVEL for original depths:
> GSPD.BDGa <- merge(x=GSPD.HOR[,c("WISE3_ID", "ID", "HOTH")],
+         y=GSPD.BDG, by=c("WISE3_ID"))
# estimate inverse distance weights:
> GSPD.BDGa$w <- 1/(GSPD.BDGa$HOTH-GSPD.BDGa$DEPTH)^2
> GSPD.BDGa$w <- ifelse(is.infinite(GSPD.BDGa$w), 0, GSPD.BDGa$w)
> GSPD.BDGa$BULKDENSa <- GSPD.BDGa$BULKDENS*GSPD.BDGa$w
> GSPD.BDGa$GRAVELa <- GSPD.BDGa$GRAVEL*GSPD.BDGa$w
# aggregate per each horizon:
> GSPD.BDG_ID <- aggregate(GSPD.BDGa[c("BULKDENSa", "GRAVELa", "w")],
+         by=list(GSPD.BDGa$ID), FUN=sum)
> GSPD.BDG_ID$BULKDENS <- GSPD.BDG_ID$BULKDENSa/GSPD.BDG_ID$w
> GSPD.BDG_ID$GRAVEL <- GSPD.BDG_ID$GRAVELa/GSPD.BDG_ID$w
> names(GSPD.BDG_ID)[1] <- "ID"
```

To combine the two tables we use:

```
> GSPD.HORa <- merge(x=GSPD.HOR[c("WISE3_ID", "HONU", "ID", "ORGC", "HOTH")],
+          y=GSPD.BDG_ID[,c("ID", "BULKDENS", "GRAVEL")], by=c("ID"))
```

and now we can estimate ORGC.d (kg C m^{-2}) using Eq.(7.2.1):

```
> GSPD.HORa$ORGC.d <- GSPD.HORa$ORGC/1000 * GSPD.HORa$HOTH/100 * GSPD.HORa$BULKDENS*1000
+             * (100-GSPD.HORa$GRAVEL)/100
> options(list(scipen=3,digits=3))
> round(summary(GSPD.HORa$ORGC.d), 1)  # total organic carbon in kg m^-2
```

[8] http://www.eoearth.org/article/soil_organic_carbon

[9] The average soil density is about 1682 kg m^{-3}. Different mean values for bulk density will apply for e.g. organic soils, Andosols, Arenosols, low activity (LAC) and high activity clays (HAC) soils.

[10] Prepared by Niels Batjes by using taxo-transfer procedures described in Batjes et al. (2007).

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    0.0     0.7     1.5     2.9     3.1   298.0  4851.0
```

where HOTH is the total thickness of the profile and ORGC.d is the estimated soil organic carbon for each horizon. We can now estimate the total soil organic carbon (SOC) in kg m^{-2} for the whole profile:

```
# select only horizons with ORGC!
> GSPD.orgc <- subset(GSPD.HORa, !is.na(GSPD.HORa$ORGC.d)&GSPD.HORa$ORGC.d>0,
+         c("WISE3_ID", "ORGC.d"))
# aggregate OGRC values per profiles (in kg / m^2):
> GSPD.orgc <- aggregate(GSPD.orgc["ORGC.d"], by=list(GSPD.orgc$WISE3_ID), FUN=sum)
# thickness of soil with biological activity:
> GSPD.hoth <- subset(GSPD.HORa, !is.na(GSPD.HORa$ORGC.d)&GSPD.HORa$ORGC.d>0,
+         c("WISE3_ID", "HOTH"))
# aggregate HOTH values to get the thickness of soil:
> GSPD.orgc$HOTH <- aggregate(GSPD.hoth["HOTH"],
+        by=list(GSPD.hoth$WISE3_ID), FUN=sum)$HOTH
```

which gives the following result:

```
> GSPD.orgc[1:10,]
```

```
    Group.1 ORGC.d HOTH
1   AF0001   4.26  150
2   AF0002  12.07  170
3   AF0003   2.70   50
4   AF0004   4.63   35
5   AF0005   3.69  190
6   AL0001  10.66   94
7   AL0002   6.31   87
8   AL0003   8.73   85
9   AL0004  22.34  120
10  AL0005  11.89  170
```

This shows that, for example, the profile AF0001 has 4.3 kg C m^{-2}, and organic carbon was observed up to the depth of 150 cm. Next, we want to obtain the coordinates of profiles:

```
# coordinates of points
> GSPD.latlon <- sqlQuery(cGSPD, query="SELECT WISE3_id, LATIT, LATDEG, LATMIN,
+     LATSEC, LONGI, LONDEG, LONMIN, LONSEC FROM WISE3_SITE")
> GSPD.latlon[1,]
```

```
   WISE3_id LATIT LATDEG LATMIN LATSEC LONGI LONDEG LONMIN LONSEC
1   AL0030     N     40     39     40     E     20     48     58
```

These need to be converted to arcdegrees i.e. merged in single column. First, we remove the missing coordinates and then convert the multiple columns to a single column:

```
# make coordinates in arcdegrees:
> GSPD.latlon <- subset(GSPD.latlon, !is.na(GSPD.latlon$LATDEG)&
+    !is.na(GSPD.latlon$LONDEG)&!is.na(GSPD.latlon$LATMIN)&
+    !is.na(GSPD.latlon$LONMIN))
> GSPD.latlon$LATSEC <- ifelse(is.na(GSPD.latlon$LATSEC), 0, GSPD.latlon$LATSEC)
> GSPD.latlon$LONSEC <- ifelse(is.na(GSPD.latlon$LONSEC), 0, GSPD.latlon$LONSEC)
# define a new function to merge the degree, min, sec columns:
> cols2dms <- function(x,y,z,e)
+    {as(char2dms(paste(x, "d", y, "'", z, "\"", e, sep="")), "numeric")}
> GSPD.latlon$LAT <- cols2dms(GSPD.latlon$LATDEG, GSPD.latlon$LATMIN,
+    GSPD.latlon$LATSEC, GSPD.latlon$LATIT)
> GSPD.latlon$LON <- cols2dms(GSPD.latlon$LONDEG, GSPD.latlon$LONMIN,
+    GSPD.latlon$LONSEC, GSPD.latlon$LONGI)
```

The two tables (horizon properties and profile locations) can be merged by using:

```
> GSPD <- merge(x=data.frame(locid=GSPD.latlon$WISE3_id, LAT=GSPD.latlon$LAT,
+     LON=GSPD.latlon$LON), y=data.frame(locid=GSPD.orgc$Group.1, HOTH=GSPD.orgc$HOTH,
+     SOC=GSPD.orgc$ORGC.d), all.y=F, all.x=T, sort=F, by.x="locid")
> str(GSPD)

  'data.frame':   8065 obs. of  5 variables:
   $ locid: Factor w/ 10253 levels "AF0001","AF0002",..: 1 2 3 4 5 6 7 8 9 10 ...
   $ LAT  : num  34.5 34.5 34.5 34.3 32.4 ...
   $ LON  : num  69.2 69.2 69.2 61.4 62.1 ...
   $ HOTH : int  150 170 50 35 190 94 87 85 120 170 ...
   $ SOC  : num  4.26 12.07 2.7 4.63 3.69 ...
```

which can be converted to a point map (Fig. 7.1), and exported to a shapefile:

```
> coordinates(GSPD) <- ~ LON+LAT
> proj4string(GSPD) <- CRS("+proj=longlat +ellps=WGS84")
# export to a shapefile:
> writeOGR(GSPD, "SOC.shp", "SOC", "ESRI Shapefile")
# plot the world distribution:
> load(url("http://spatial-analyst.net/book/system/files/worldborders.RData"))
> bubble(subset(GSPD, !is.na(GSPD$SOC))["SOC"], col="black",
+      sp.layout=list("sp.lines", worldborders, col="light grey"))
```

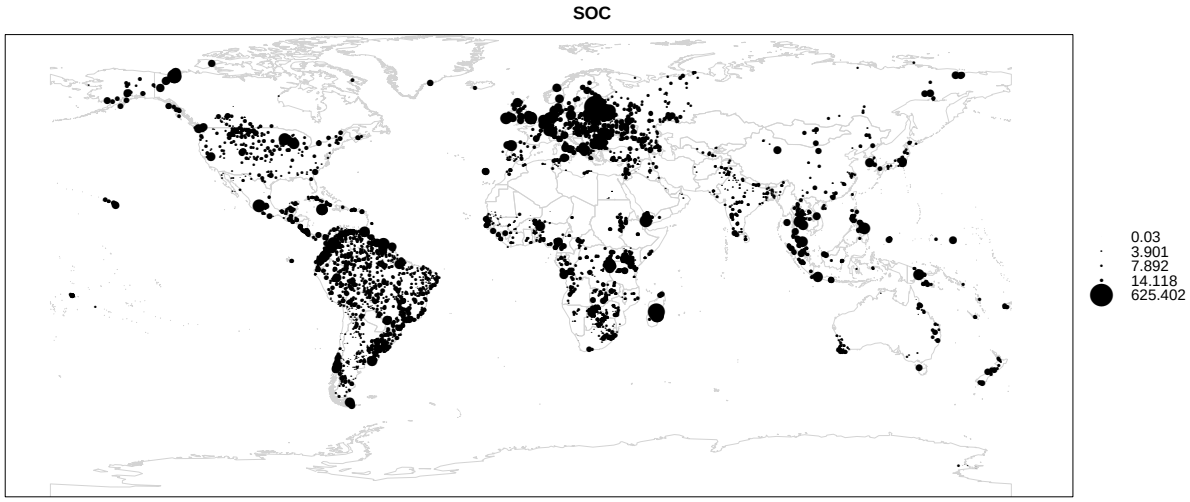

Fig. 7.1: Global distribution of soil profiles in the ISRIC WISE v3 database and values of total soil organic carbon (SOC in kg C m^{-2}). Note that the distribution of points is highly non-uniform — many large areas are not represented. See Batjes (2008) for more info.

Because we will use SAGA GIS to run the interpolation, you will also need to convert the downloaded worldmaps to the SAGA grid format:

```
> rsaga.esri.to.sgrd(in.grids=set.file.extension(map.list, ".asc"),
+      out.sgrds=set.file.extension(map.list, ".sgrd"), in.path=getwd())
```

Have in mind that these are relatively large grids (3600×1200 pixels), so the conversion process can take few minutes. To check that conversion was successful, you can open the maps in SAGA.

7.3 Regression modeling

Now that we have prepared a point map showing values of aggregated target variables, we can overlay the points over predictors (worldmaps) and prepare a regression matrix. Because there are many maps and they are relatively large, we run this operation instead using SAGA GIS[11]:

```
> rsaga.geoprocessor(lib="shapes_grid", module=0, param=list(SHAPES="SOC.shp",
+     GRIDS=paste(set.file.extension(map.list, ".sgrd"), collapse=";"),
+     RESULT="SOC_ov.shp", INTERPOL=0))  # simple nearest neighbor overlay

  SAGA CMD 2.0.4

  library path:   C:/PROGRA~2/R/R-29~1.2/library/RSAGA/saga_vc/modules
  library name:   shapes_grid
  module name :   Add Grid Values to Points
  author      :   (c) 2003 by O.Conrad

  Load shapes: SOC.shp...
  ready

  Load grid: biocl01.sgrd...
  ready

  ...

  Points: GSPD
  Grids: 25 objects (biocl01, biocl02, biocl04, biocl05, biocl06, biocl12, biocl15,
   countries, dcoast, globedem, landcov, landmask, nlights, pcndvi1, pcndvi2,
   pcndvi3, pcpopd1, himpact, glwd31, wildness, gcarb, quakein, iflworld, treecov))
  Result: Result
  Interpolation: Nearest Neighbor

  ready
  Save shapes: SOC_ov.shp...

  ready
  Save table: SOC_ov.dbf...
```

This will produce a point shapefile, which we can then read back into R:

```
> SOC.ov <- readShapePoints("SOC_ov.shp", CRS("+proj=longlat +ellps=WGS84"))
# fix the names:
> names(SOC.ov@data)[4:length(SOC.ov@data)] <- map.list
# note that SAGA can not generate NA values but inserts instead "0" values!!
> SOC.ov <- subset(SOC.ov, SOC.ov$landmask==1&SOC.ov$HOTH>0)
# some points fall outside the landmask!
> str(SOC.ov@data)

  'data.frame':   7681 obs. of  30 variables:
   $ LOCID    : Factor w/ 8065 levels "AF0001","AF0002",..: 1 2 3 4 5 6 7 8 9 10 ...
   $ HOTH     : int  150 170 50 35 190 94 87 85 120 170 ...
   $ SOC      : num  4.26 12.07 2.7 4.63 3.69 ...
   $ biocl01  : num  119 119 119 172 199 104 138 110 160 156 ...
   $ biocl02  : num  149 149 149 156 174 ...
   $ biocl04  : num  8678 8678 8678 8549 9028 ...
   $ biocl05  : num  321 321 321 375 421 268 294 270 291 291 ...
   $ biocl06  : num  -82 -82 -82 -13 -9 -28 16 -30 48 40 ...
   $ biocl12  : num  340 340 340 222 79 ...
   $ biocl15  : num  98 98 98 102 107 32 61 26 46 44 ...
   $ countries: num  1 1 1 1 1 2 85 2 2 2 ...
```

[11] Loading such a large quantity of maps to R would be very inefficient and is not recommended for OS with <4GB RAM.

```
$ dcoast   : num  1091 1091 1091 803 782 ...
$ globedem : num  1790 1790 1790 776 780 ...
$ landcov  : num  9 9 9 9 9 4 1 11 11 11 ...
$ landmask : num  1 1 1 1 1 1 1 1 1 1 ...
$ nlights  : num  3 3 3 0 0 6 0 0 4 5 ...
$ pcndvi1  : num  2295 2295 2295 2140 2238 ...
$ pcndvi2  : num  -77 -77 -77 -178 -159 -64 -244 13 -170 -153 ...
$ pcndvi3  : num  123 123 123 118 122 108 127 73 152 128 ...
$ pcpopd1  : num  30076.6 30076.6 30076.6 24.5 81.9 ...
$ himpact  : num  1 1 1 1 1 1 1 1 1 1 ...
$ glwd31   : num  0 0 0 0 0 0 0 0 0 0 ...
$ wildness : num  0 0 0 0 0 0 0 0 0 0 ...
$ hwsd     : num  10 10 10 10 7 21 20 26 4 26 ...
$ gcarb    : num  6.5 6.5 6.5 4.2 2.5 ...
$ quakein  : num  7.9 7.9 7.9 3.2 0 ...
$ iflworld : num  0 0 0 0 0 0 0 0 0 0 ...
$ treecov  : num  0 0 0 0 0 0 0 0 0 0 ...
$ coords.x1: num  69.2 69.2 69.2 61.4 62.1 ...
$ coords.x2: num  34.5 34.5 34.5 34.3 32.4 ...
```

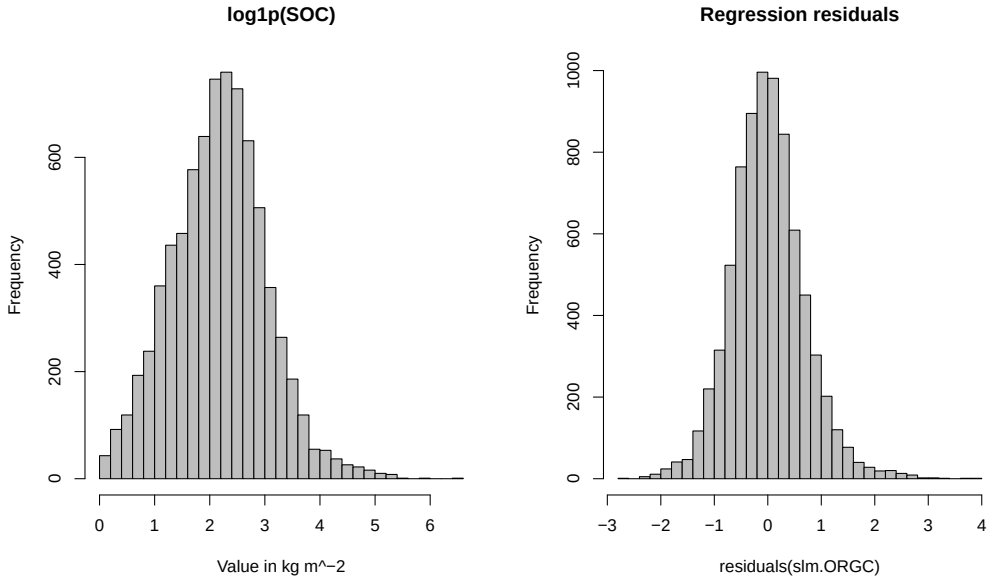

Fig. 7.2: Target variable (soil organic carbon) after the necessary transformation and histogram for the regression residuals.

Before we proceed with regression analysis, it is a good idea to visualize histograms for the target variable, in order to see if they need to be transformed before model fitting[12]. You will soon notice that SOC needs to be transformed before regression modeling (Fig. 7.2):

```
> hist(log1p(SOC.ov$SOC), col="grey")
```

The transformed variable shows close to normal distribution, so that we can now fit a regression model:

```
> orgc.formula <- as.formula(paste("log1p(SOC)~", paste(sub(".asc", "",
+    map.list[!(map.list %in% c("landmask", "countries", "wwfeco"))]), collapse="+")))
# some maps we do not need!
> orgc.formula
```

[12]Close-to-normal distribution is a prerequisite for regression modeling.

```
  log1p(SOC) ~ biocl01 + biocl02 + biocl04 + biocl05 + biocl06 +
      biocl12 + biocl15 + dcoast + globedem + landcov + nlights +
      pcndvi1 + pcndvi2 + pcndvi3 + pcpopd1 + himpact + glwd31 +
      wildness + hwsd + gcarb + quakein + iflworld + treecov
```

```
> lm.ORGC <- lm(orgc.formula, SOC.ov@data)
> slm.ORGC <- step(lm.ORGC, trace=-1) # step-wise regression
> summary(slm.ORGC)$adj.r.squared
```

```
  [1] 0.363
```

This shows that the predictors explain 36% of variability in the SOC values (cumulative density of organic carbon in the soil). For Digital Soil Mapping projects (Lagacherie et al., 2006), this is a promising value.

For practical reasons (computational intensity), we will further focus on using only the top 20 most significant predictors to generate predictions. These can be selected by using:

```
> pr.rank <- rank(summary(slm.ORGC)$coefficients[,4])<20
> SOC.predictors <- attr(summary(slm.ORGC)$coefficients[pr.rank,1], "names")[-1]
> SOC.predictors
```

```
   [1] "biocl01"   "biocl02"   "biocl04"   "biocl12"
   [5] "globedem"  "landcov9"  "landcov12" "nlights"
   [9] "glwd312"   "glwd314"   "glwd317"   "hwsd4"
  [13] "hwsd6"     "hwsd17"    "hwsd19"    "hwsd25"
  [17] "hwsd26"    "quakein"
```

After we have determined the top 20 most significant predictors, we can make predictions by using the SAGA **multiple linear regression module**. However, before we can produce predictions in SAGA, we need to prepare the indicator maps and a shapefile with transformed target variable. For example, to prepare indicators for different classes of land cover, we can use:

```
> for(j in c("9","12")){
>   worldmaps$tmp <- ifelse(worldmaps$landcov==j, 1, 0)
>   write.asciigrid(worldmaps["tmp"], paste("landcov", j, ".asc", sep=""), na.value=-1)
> }
...
# list all indicators and convert to SAGA grids:
> indicator.grids <- c(list.files(getwd(), pattern="hwsd[[:digit:]]*.asc",
+         recursive=F, full=F),
+         list.files(getwd(), pattern="glwd31[[:digit:]]*.asc", recursive=F, full=F),
+         list.files(getwd(), pattern="landcov[[:digit:]]*.asc", recursive=F, full=F))
> rsaga.esri.to.sgrd(in.grids=indicator.grids,
+         out.sgrds=set.file.extension(indicator.grids, ".sgrd"), in.path=getwd())
```

We also need to prepare the point map with transformed target variables:

```
> SOC.ov$SOC.T <- log1p(SOC.ov$SOC)
> SOC.ov$HOTH.T <- sqrt(SOC.ov$HOTH)
> writeOGR(SOC.ov[c("SOC.T","HOTH.T")], "SOC_ov.shp", "SOC_ov", "ESRI Shapefile")
```

which now allows us to use SAGA GIS to make predictions using multiple linear regression:

```
> rsaga.geoprocessor(lib="geostatistics_grid", module=4,
+    param=list(GRIDS=paste(set.file.extension(SOC.predictors, ".sgrd"), collapse=";"),
+    SHAPES="SOC_ov.shp", ATTRIBUTE=0, TABLE="regout.dbf", RESIDUAL="res_SOC.shp",
+    REGRESSION="SOC_reg.sgrd", INTERPOL=0))
```

```
  ...
  1: RÂš = 15.508441% [15.508441%] -> biocl02

  2: RÂš = 21.699108% [6.190666%] -> globedem
```

```
3: RÂš = 24.552229% [2.853121%] -> hwsd4

4: RÂš = 26.552463% [2.000235%] -> biocl12

5: RÂš = 30.908089% [4.355626%] -> biocl01

6: RÂš = 31.498327% [0.590238%] -> hwsd26

7: RÂš = 31.993559% [0.495233%] -> hwsd6

8: RÂš = 32.407088% [0.413529%] -> hwsd17

9: RÂš = 32.738160% [0.331072%] -> landcov12

10: RÂš = 33.136920% [0.398761%] -> landcov9

11: RÂš = 33.434208% [0.297288%] -> hwsd19

12: RÂš = 33.700079% [0.265871%] -> biocl04

...
```

which shows that the best predictors are Mean Diurnal Range (biocl02), elevation (globedem), various soil types (hwsd), annual temperature (biocl01), temperature seasonality (biocl04), annual precipitation (biocl12), and land cover classes. Most of variation in SOC values can be explained by using only bioclimatic maps.

The resulting map (Fig. 7.3) shows that high organic carbon concentration in soil can be mainly observed in the wet and cooler areas (mountain chains); deserts and areas of low biomass have distinctly lower soil organic carbon. Surprisingly, the model predicts high SOC concentration also in arctic zones (Greenland) and Himalayas, which is an obvious artifact. Recall that the sampling locations have not been chosen to represent all possible environmental conditions, so the model is probably extrapolating in these areas (see also p.59).

To speed up further analysis we will focus on estimating SOC for South American continent only. This is the continent with best (most consistent) spatial coverage, as visible from Fig. 7.3 (below). For further analysis, we do not really need all maps, but just the estimate of the trend (SOC_reg). We can reproject the maps using (see also §6.5):

```
> SA.aea <- "+proj=aea +lat_1=-5 +lat_2=-42 +lat_0=-32 +lon_0=-60 +x_0=0 +y_0=0
+      +ellps=aust_SA +units=m +no_defs"
> rsaga.geoprocessor(lib="pj_proj4", 2,
+      param=list(SOURCE_PROJ="\"+proj=longlat +datum=WGS84\"",
+      TARGET_PROJ=paste('"', SA.aea ,'"', sep=""), SOURCE="SOC_reg.sgrd",
+      TARGET="m_SOC_reg.sgrd", TARGET_TYPE=2, INTERPOLATION=1,
+      GET_SYSTEM_SYSTEM_NX=586, GET_SYSTEM_SYSTEM_NY=770, GET_SYSTEM_SYSTEM_X=-2927000,
+      GET_SYSTEM_SYSTEM_Y=-2597000, GET_SYSTEM_SYSTEM_D=10000))

SAGA CMD 2.0.4

library path:   C:/PROGRA~2/R/R-29~1.2/library/RSAGA/saga_vc/modules
library name:   pj_proj4
module name :   Proj.4 (Command Line Arguments, Grid)
author      :   O. Conrad (c) 2004-8

Load grid: SOC_reg.sgrd...
ready

Parameters

Inverse: no
Source Projection Parameters: +proj=longlat +datum=WGS84
Target Projection Parameters: +proj=aea +lat_1=-5 +lat_2=-42 +lat_0=-32
+lon_0=-60 +units=m +no_defs +x_0=0 +y_0=0 +ellps=aust_SA
```

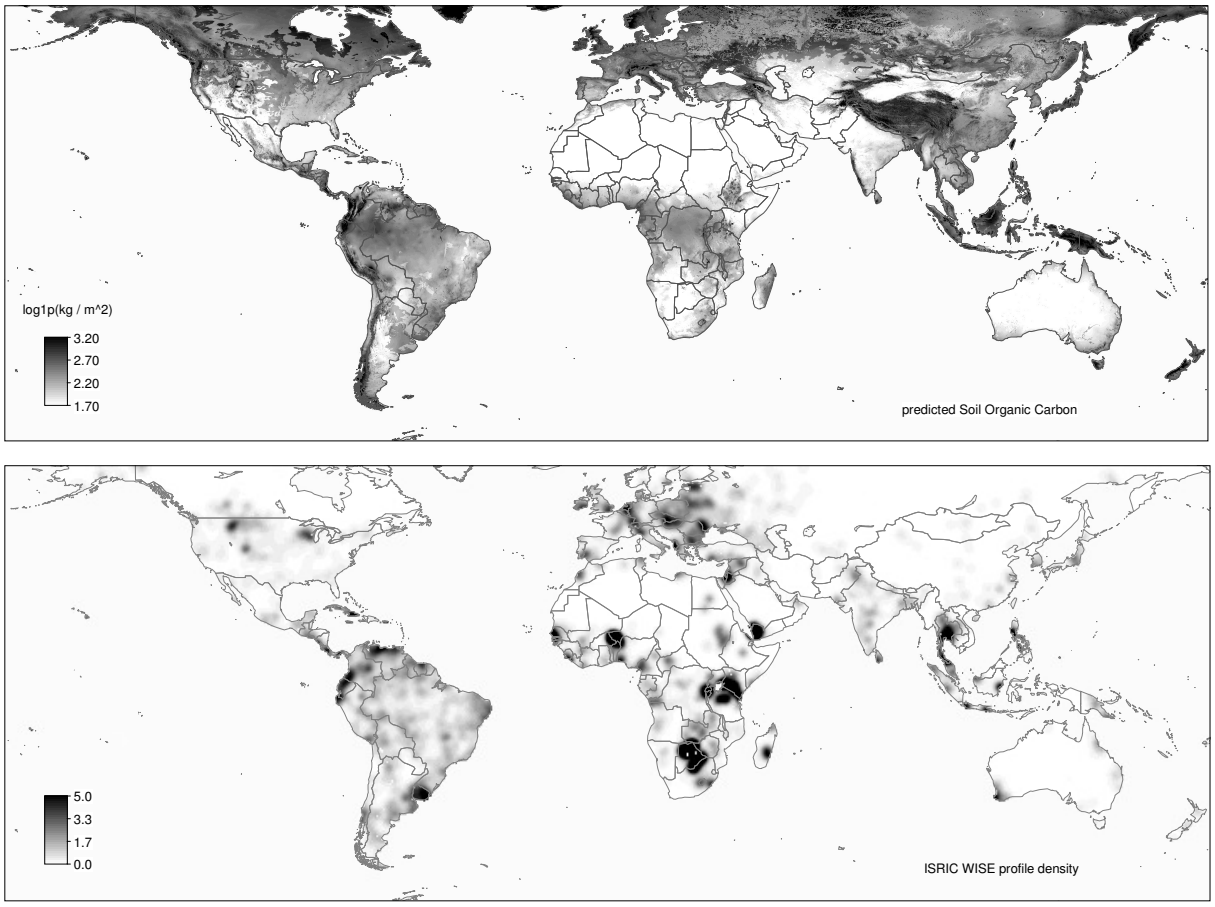

Fig. 7.3: Predicted values of the target variable (`log1p(SOC)`) using the 20 most significant predictors and multiple linear regression module in SAGA GIS (above). ISRIC WISE coverage map — sampling density on 0.5 arcdegree grid derived using kernel smoothing (below).

```
  Grid system: 0.1; 3600x 1300y; -179.95x -64.95y
  Source: SOC_reg
  Target: [not set]
  Shapes: [not set]
  X Coordinates: [not set]
  Y Coordinates: [not set]
  Create X/Y Grids: no
  Target: grid system
  Interpolation: Bilinear Interpolation

  Source: +proj=longlat +datum=WGS84

  Target: +proj=aea +lat_1=-5 +lat_2=-42 +lat_0=-32 +lon_0=-60 +x_0=0
  +y_0=0 +ellps=aust_SA +units=m +no_defs

  Save grid: m_SOC_reg.sgrd...
  ready

> gridsSA <- readGDAL("m_SOC_reg.asc")

  m_SOC_reg.asc has GDAL driver AAIGrid
  and has 770 rows and 586 columns
```

```
> names(gridsSA) <- "SOC_reg"
> proj4string(gridsSA) <- CRS(SA.aea)
> SA.bbox <- gridsSA@bbox
> SA.bbox

        min     max
x -2932000 2928000
y -2602000 5098000
```

which will reproject and resample the predicted `log1p(SOC)` map from geographic coordinates to the Albers Equal-Area Conic projection system[13], commonly used to represent the whole South American continent.

7.4 Modeling spatial auto-correlation

We have explained some 36% of variation in the `SOC` values using worldmaps. Next we can look at the variograms i.e. try to improve interpolations using kriging. Because we focus only on the South American continent, we also need to subset the point map of profiles:

```
# reproject the profile data:
> GSPD.aea <- spTransform(GSPD, CRS(SA.aea))
> writeOGR(GSPD.aea, "GSPD_aea.shp", ".", "ESRI Shapefile")
# subset the points:
> rsaga.geoprocessor(lib="shapes_tools", module=14,
+      param=list(SHAPES="GSPD_aea.shp", CUT="m_GSPD_aea.shp",
+      METHOD=0, TARGET=0, CUT_AX=SA.bbox[1,1], CUT_BX=SA.bbox[1,2],
+      CUT_AY=SA.bbox[2,1], CUT_BY=SA.bbox[2,2]))
```

which will subset the input point map to 1729 points. These can be now analyzed for spatial auto-correlation:

```
> m_SOC <- readShapePoints("m_GSPD_aea.shp", CRS(SA.aea))
> m_SOC.ov <- overlay(gridsSA, m_SOC)
> m_SOC.ov$SOC <- m_SOC$SOC
> m_SOC.ov <- remove.duplicates(m_SOC.ov)  # many duplicate points!
> sel <- !is.na(m_SOC.ov$SOC)&!is.na(m_SOC.ov$SOC_reg)
> res_SOC.svar <- variogram(log1p(SOC) ~ SOC_reg, m_SOC.ov[sel,])
> SOC.rvgm <- fit.variogram(res_SOC.svar, vgm(nugget=var(SOC.ov$SOC, na.rm=T)/2,
+     model="Exp", range=80000, sill=var(SOC.ov$SOC, na.rm=T)/2))
> SOC.rvgm

  model     psill    range
1   Nug 0.3879486      0.0
2   Exp 0.1151655 823650.5
```

which shows that the residuals are correlated up to the distance of >1000 km. This number seems unrealistic. In practice, we know that soils form mainly at watershed level or even at short distances, so chances that two profile locations that are so far away still make influence on each other are low. On the other hand, from statistical perspective, there is no reason not to utilize this auto-correlation to improve the existing predictions.

Variograms for the original variable and regression residuals can be seen in Fig. 7.4. Note also that the variance of the residuals is about 70% of the original variance, which corresponds to the R-square estimated by the regression model. Compare also this plot to some previous exercises, e.g. Fig. 5.8. We can in general say that the nugget variation of `SOC` is relatively high, which indicates that our estimate of global `SOC` will be of limited accuracy.

7.5 Adjusting final predictions using empirical maps

Once we have estimated the variogram for residuals, we can proceed with regression-kriging[14]:

[13] `http://spatialreference.org/ref/esri/102033/`

[14] In this case implemented as kriging with external drift, and with a single predictor — regression estimate (see §2.1.4).

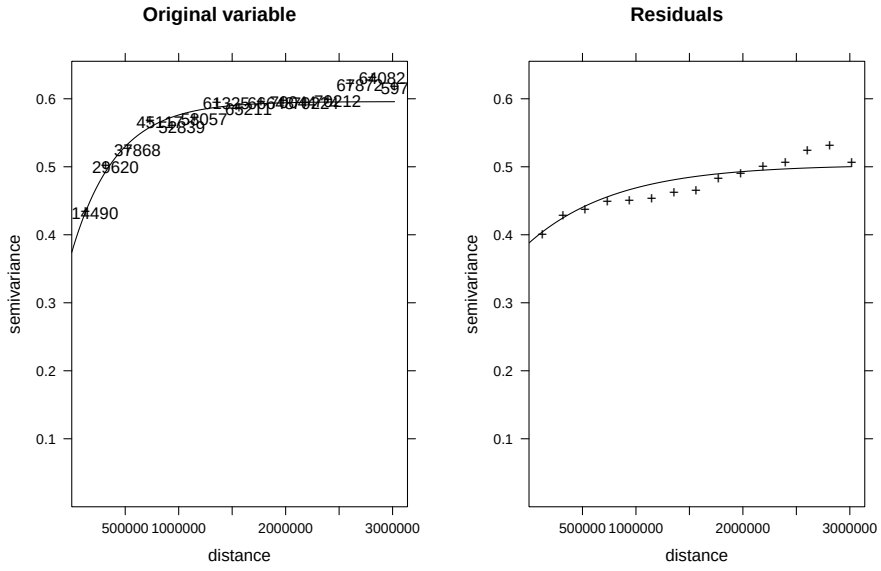

Fig. 7.4: Variograms for SOC fitted in gstat using standard settings.

```
# block regression-kriging:
> m_SOC.rk <- krige(log1p(SOC) ~ SOC_reg, m_SOC.ov[sel,], gridsSA, SOC.rvgm,
+      nmin=30, nmax=40, block=c(10e3, 10e3))

  [using universal kriging]
  Warning message:
  In points2grid(points, tolerance, round, fuzz.tol) :
    grid has empty column/rows in dimension 2


# back-transform values:
> m_SOC.rk$SOC_rk <- expm1(m_SOC.rk$var1.pred)
```

in this case, gstat reported about empty pixels in the map that have been removed. The final regression-kriging map of SOC for South American continent can be seen in Fig. 7.5. The RK model predicts even in the areas where there are almost no soil profiles, hence the map is possibly of poor quality in some regions. To improve this map, we can use the USDA–produced Soil Organic Carbon Map[15], which is shown in Fig. 7.5 (2):

```
# reproject and import the USDA map:
> download.file("http://spatial-analyst.net/worldmaps/SOC.zip",
+     destfile=paste(getwd(), "/SOC.zip", sep=""))
> unzip("SOC.zip")
# resample to the same grid:
> rsaga.esri.to.sgrd(in.grids="SOC.asc", out.sgrd="SOC.sgrd", in.path=getwd())
> rsaga.geoprocessor(lib="pj_proj4", 2,
+     param=list(SOURCE_PROJ="\"+proj=longlat +datum=WGS84\"",
+     TARGET_PROJ=paste('"', SA.aea ,'"', sep=""),
+     SOURCE="SOC.sgrd", TARGET="m_SOC_USDA.sgrd", TARGET_TYPE=2, INTERPOLATION=1,
+     GET_SYSTEM_SYSTEM_NX=m_SOC.rk@grid@cells.dim[[1]],
+     GET_SYSTEM_SYSTEM_NY=m_SOC.rk@grid@cells.dim[[2]],
+     GET_SYSTEM_SYSTEM_X=m_SOC.rk@grid@cellcentre.offset[[1]],
+     GET_SYSTEM_SYSTEM_Y=m_SOC.rk@grid@cellcentre.offset[[2]],
+     GET_SYSTEM_SYSTEM_D=10000))
> rsaga.sgrd.to.esri(in.sgrds="m_SOC_USDA.sgrd", out.grids="m_SOC_USDA.asc",
+     out.path=getwd(), prec=3)
> m_SOC.rk$SOC_USDA <- readGDAL("m_SOC_USDA.asc")$band1
```

[15] http://soils.usda.gov/use/worldsoils/mapindex/

Recall from §2.1.3 that, if we know the uncertainty of both maps, we can derive a weighted average and create a combined prediction. In this case, we do not have any estimate of the uncertainty of the USDA SOC map; we only have an estimate of the uncertainty of RK SOC map. Because there are only two maps, the weights for the RK map (Fig. 7.5 (3)) can be derived using the inverse of the relative prediction variance (see p.25); the remaining weights for the USDA map can be derived as $1 - w$. Or in R syntax:

```
# merge the two maps (BCSP formula):
> w <- sqrt(m_SOC.rk$var1.var)/sqrt(var(log1p(m_SOC.ov$SOC), na.rm=T))
> m_SOC.rk$w <- 1-w/max(w, na.rm=TRUE)
> m_SOC.rk$SOC.f <- m_SOC.rk$w * m_SOC.rk$SOC_rk + (1-m_SOC.rk$w) * m_SOC.rk$SOC_USDA
```

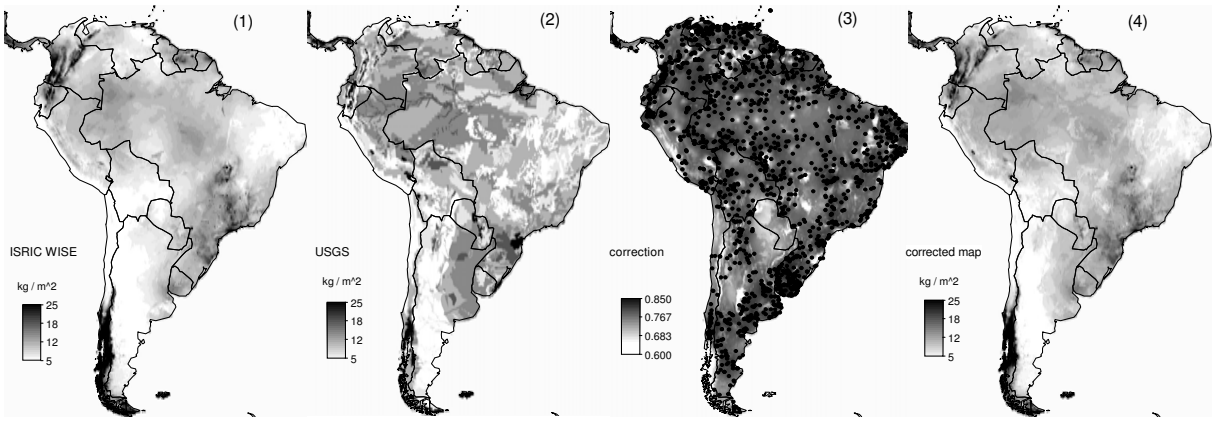

Fig. 7.5: Soil Organic Carbon stock (kg C m^{-2}) for South America: (1) predicted using regression-kriging, (2) the USDA SOC map produced using soil regions; (3) sampling locations and map of weights derived as the inverse relative prediction error; (4) the final corrected map of SOC derived as a weighted average between the maps (1) and (2).

The final corrected map of SOC is shown in Fig. 7.5 (4). In this case, the USDA map is assume to be more spatially 'consistent' about the actual SOC stock. The weighted average between the two maps is possibly the best estimate of the Soil Organic Carbon given the limited data. To validate this map, one would need to collect block estimates of SOC with a support size of 10 km (Heuvelink and Pebesma, 1999).

7.6 Summary points

Estimation of organic carbon stock using ISRIC WISE profiles and geostatistical techniques is possible, but the final map is of limited quality: (a) soil samples are fairly clustered (Fig. 7.3, below), for many regions there are still no measured soil data; (b) predictors used are rather coarse (cca. 10 km), which limits the regression modeling; (c) the residuals for SOC consequently show high nugget. The map presented in Fig. 7.5 (1) can be considered to be especially poor where the density of point samples is low. The question remains whether the models would improve if one would consider fitting variogram models locally (moving window), or by using finer-grain predictors (<10 km) that could potentially be able to explain short-range variation. On the other hand, we know that there is inherent uncertainty in the geo-locations of WISE profiles, so that not even finer-grain predictors would help us improve the predictions.

If you repeat a similar analysis with other soil variables of interest, you will notice that the gridded predictors explain only between 10–40% of the observed variability in the values (e.g. 36% for SOC, 11% for HOTH, 22% for SAND, 28% for SILT, 15% for CLAY), which means that these maps are of limited accuracy. The variograms also show relatively high nugget, which also means that about 30–60% of variability in these target variables cannot be explained by regression-kriging. This is particulary problematic for large regions that are completely under represented — most of the former Russian federation, Australia and Canada (see the map in Fig. 7.1). Nevertheless, the main patterns of soil parameters produced using ISRIC WISE will often correspond to our empirical knowledge: high soil organic carbon mainly reflects the cold and wet temperatures (Batjes, 1996); deep soils are predicted in the tropical regions and areas of high biomass (Eswaran et al., 1993); texture classes are connected with the land cover, relief and soil mapping units etc.

The advantage of automating the operations, on the other hand, is that these maps can be easily updated once the ISRIC WISE becomes more representative and of higher quality. Due to the data processing automation, many other important soil parameters from the ISRIC WISE database could easily be revised once updates with better geographical coverage are released .

Self-study exercises:

(1.) Estimate nugget variation for `SOC` for the five largest countries in the world. Plot the variograms one over the other.

(2.) Compare the Global Biomass Carbon Map distributed by The Carbon Dioxide Information Analysis Center and the total soil carbon map shown in Fig. 7.5(4). Are the two maps correlated and how much? Where is the difference highest and why?

(3.) Repeat the spatial prediction of soil carbon by focusing on the North American continent (HINT: resample the maps following the previous exercise in §6.5.)

(4.) Which country in the world has highest reserves of organic carbon in absolute terms (total soil carbon in tones), and which one in relative terms (average density of carbon)?

(5.) Compare spatial prediction of `SOC` for South America and Africa (regression-kriging variance). Why are soil profile data in South America more suited for geostatistical mapping?

(6.) Interpolate soil textures (`SAND`, `SILT`, `CLAY`) using the same procedure explained in the text and produce global maps.

(7.) Focus on Australia and compare the soil organic carbon map available from the Australian soil atlas with the map shown in Fig. 7.3. Plot the two maps next to each other using the same grid settings.

Further reading:

★ Batjes, N.H., 2009. Harmonized soil profile data for applications at global and continental scales: updates to the WISE database. Soil Use and Management 25, 124-127.

★ Lagacherie, P., McBratney, A.B., Voltz, M., (eds) 2006. **Digital Soil Mapping: An Introductory Perspective**. Developments in Soil Science, Volume 31. Elsevier, Amsterdam, 350 p.

★ Ruesch, A., Gibbs, H.K., 2008. New IPCC Tier-1 Global Biomass Carbon Map For the Year 2000. Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 45 p.

★ `http://www.isric.org` — ISRIC World Soil Information center;

★ `http://www.pedometrics.org` — The international research group on pedometrics;

★ `http://www.globalsoilmap.net` — International consortium that aims to make a new digital soil map of the world using state-of-the-art and emerging technologies for soil mapping and predicting soil properties at fine resolution;