Food and Agriculture Organization of the United Nations

AFACI Asian Food & Agriculture Cooperation Initiative

Rural Development Administration

EduSoils

Regression Kriging

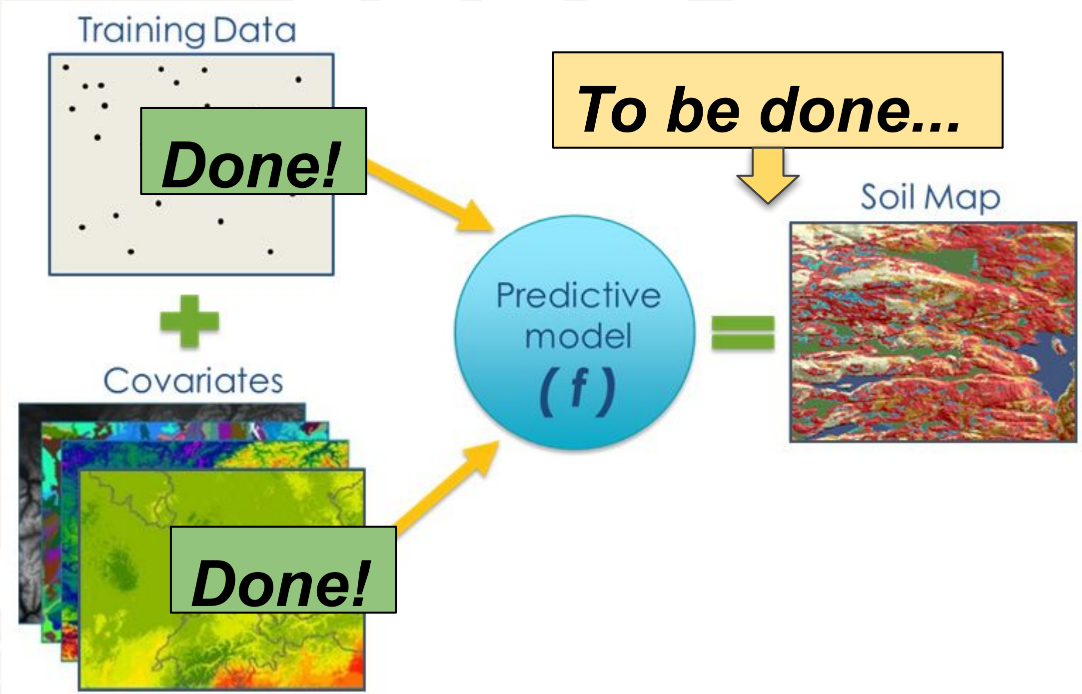
Linear modelling

GLOBAL SOIL PARTNERSHIP

Table of contents:

- Predictive modelling
- Regression Kriging
- Extracting covariate values
- Categorical variables in modelling
- Linear regression
- Assumptions of linear regression
- Multiple Linear Regression in R
- Stepwise variable selection
- Graphical diagnosis of MLR
- Collinearity test, Bonferroni test
- Mapping OCS using the MLR model

Where we stand?



Predictive modelling

In this course we present 2 most popular methods of modelling soil properties in DSM:

1) **Regression Kriging** is a hybrid model with 2 components:
    - deterministic component - multiple linear regression;
    - stochastic component - kriging;
1) **Random forest** is machine learning algorithm that uses a different combination of prediction factors to train multiple regression trees.

We present both methods for comparison, because there is **no best mapping method** for DSM, and testing and selection has to be done for every data scenario.

Let’s get started!

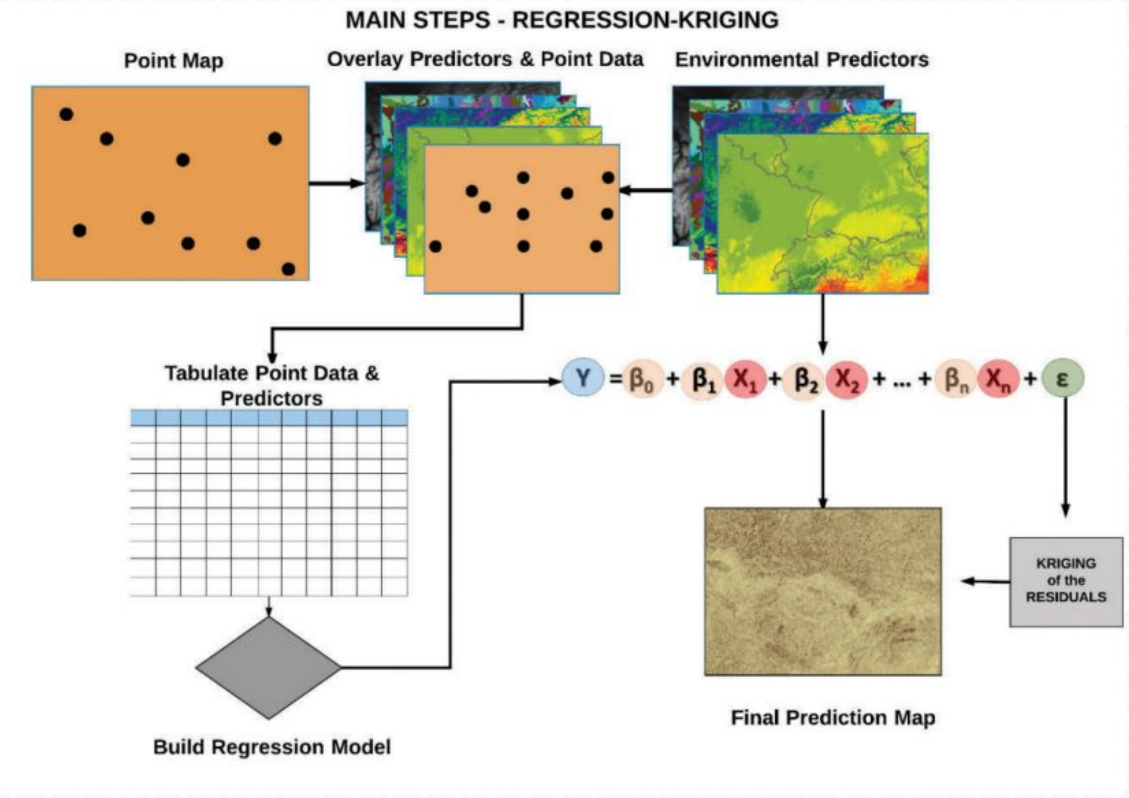
```r
# Set working directory
setwd("C:/Training Indonesia/Macedonia")

# Load the covariates stack.
load(file = "02-Outputs/covariates.RData")
names(covs)

# Load the processed data for digital soil mapping.
dat <- read.csv("02-Outputs/dat_train.csv")
names(dat)
```

Regression Kriging

Regression Kriging is a spatial interpolation technique that combines a **regression** of the **dependent variable** on **predictors** (i.e. the environmental covariates) with **kriging** of the prediction **residuals**.

Steps of Regression Kriging:

1. **Overlaying** the point data with the dependent variable (e.g. OCS) and environmental covariates (predictors). **Extracting** covariate data.
2. Fitting **multiple regression model** using the table that contains data from dependent variable and predictors.
3. In particular cases, **stepwise** multiple linear regression (MLR) can be used to eliminate insignificant predictors.
4. **Kriging** of the residuals (prediction errors): the regression model produces the **residuals** which we need to krige and add to the model predictions.


MAIN STEPS - REGRESSION-KRIGING
Point Map
Overlay Predictors & Point Data
Environmental Predictors
Tabulate Point Data & Predictors
Y = β0 + β1 X1 + β2 X2 + ... + βn Xn + ε
KRIGING of the RESIDUALS
Final Prediction Map
Build Regression Model

Extracting covariate values

```
library(sp)

class(dat)
```

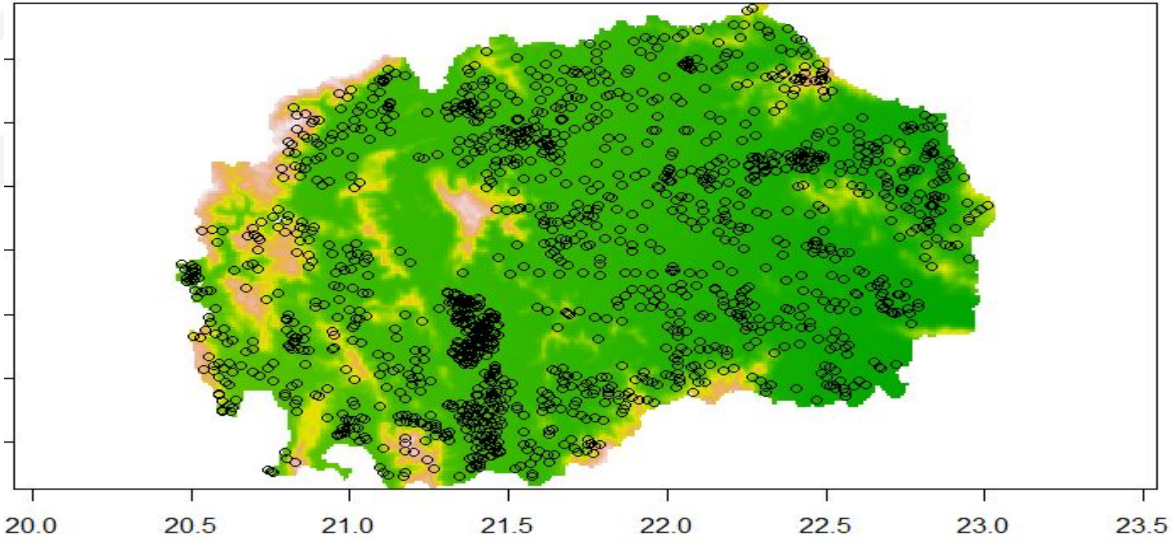
```
> class(dat)
[1] "data.frame"
```

```
# Promote to spatialPointsDataFrame and set the coordinate system
coordinates(dat) <- ~ X + Y
proj4string(dat) = CRS("+init=epsg:4326") # WGS84
class(dat)
```

```
> class(dat)
[1] "SpatialPointsDataFrame"
attr(,"package")
[1] "sp"
```

Extracting covariate values

```
# Check if the points overlay with covariates

plot(covs$B04CHE3)

points(dat)
```



Extracting covariate values

```r
# extract values from covariates to the soil points

dat <- extract(x = covs, y = dat, sp = TRUE)

summary(dat)
```

```
   LandCover          soilmap
 Min.   : 11.00   Min.   : 1.000
 1st Qu.: 20.00   1st Qu.: 5.000
 Median : 20.00   Median : 8.000
 Mean   : 43.76   Mean   : 9.108
 3rd Qu.: 50.00   3rd Qu.:13.000
 Max.   :210.00   Max.   :19.000
                  NA's   :14
```

```r
# Remove NA values

dat<-as.data.frame(dat)

dat <- dat[complete.cases(dat),]
```

Categorical variables in modelling

<u>Continuous variables</u>

- Represent **values** (e.g. carbon content, elevation, temperature)
- Data type in R: **numeric**
- **Can** be used for arithmetical operations (+,-,*,/, etc.)

Numeric logic:

  - 2 = 2 : TRUE
  - 1 != 3 : TRUE
  - 1 < 2 : TRUE
  - 3 > 1 : TRUE
  - 3 - 2 = 1 : TRUE

<u>Categorical variables</u>

- Represent **classes** (e.g. soil types, land cover classes)
- Data type in R: **factor**
- **Cannot be** used for arithmetical operations (+,-,*,/, etc.)

Factor logic:

  - 2 = 2 : TRUE
  - 1 != 3 : TRUE
  - 1 < 2 : FALSE
  - 3 > 1 : FALSE
  - 3 - 2 = 1 : FALSE

Categorical variables in modelling

1	1	3	3
3	2	1	3
2	2	2	1
2	2	2	3

est **2 - Cropland** **3 - Grassland**

Conversion to binary layers (dummies)

1	1	0	0
0	0	1	0
0	0	0	1
0	0	0	0

1 - Forest, 0 - Not forest

0	0	0	0
0	1	0	0
1	1	1	0
1	1	1	0

1 - Cropland, 0 - Not cropland

0	0	1	1
1	0	0	1
0	0	0	0
0	0	0	1

1 - Grassland, 0 - Not grassland

Checking data types

```
str(dat)
```

```
'data.frame':   1455 obs. of  20 variables:
 $ id       : Factor w/ 1469 levels "P0007","P0008"
 $ Y        : num  42 42.1 42 42 42 ...
 $ X        : num  20.8 20.8 20.9 20.9 20.9 ...
 $ SOC      : num  3.42 2.3 2.29 4.29 4.49 ...
 $ BLD      : num  1.12 1.28 1.28 1.04 1.02 ...
 $ CRFVOL   : num  22 31.9 27.3 26 16.4 ...
 $ OCS      : num  89.8 60.1 63.8 98.7 114.4 ...
 $ OCSlog   : num  4.5 4.1 4.16 4.59 4.74 ...
 $ B04CHE3  : num  553 693 672 616 638 ...
 $ PRSCHE3  : num  1053 780 952 974 927 ...
 $ TMDMOD3  : num  280 285 287 286 287 281 286 288 2
 $ DEMENV5  : num  2207 1243 1492 1809 1731 ...
 $ B07CHE3  : num  37.8 42.1 41.5 39.8 40.4 ...
 $ HIST     : num  1 1 1 1.02 1.29 ...
 $ B13CHE3  : num  125 99.8 133.8 136.6 134.4 ...
 $ B14CHE3  : num  60.3 42.4 47 47.9 43.3 ...
 $ REDL00   : num  19 22 29 53 56 21 52 22 12 17 ...
 $ TWIMRG5  : num  62 81 72 67 65 60 59 61 65 60 ...
 $ LandCover: num  20 20 50 20 20 20 30 20 50 50 ...
 $ soilmap  : num  13 13 13 13 13 13 13 2 2 2 ...
```

```
dat$LandCover <- as.factor(dat$LandCover)

dat$soilmap <- as.factor(dat$soilmap)
```

Saving the regression matrix

```
str(dat)
```

```
'data.frame':   1455 obs. of  20 variables:
 $ id        : Factor w/ 1469 levels "P0007","P0008",..
 $ Y         : num  42 42.1 42 42 42 ...
 $ X         : num  20.8 20.8 20.9 20.9 20.9 ...
 $ SOC       : num  3.42 2.3 2.29 4.29 4.49 ...
 $ BLD       : num  1.12 1.28 1.28 1.04 1.02 ...
 $ CRFVOL    : num  22 31.9 27.3 26 16.4 ...
 $ OCS       : num  89.8 60.1 63.8 98.7 114.4 ...
 $ OCSlog    : num  4.5 4.1 4.16 4.59 4.74 ...
 $ B04CHE3   : num  553 693 672 616 638 ...
 $ PRSCHE3   : num  1053 780 952 974 927 ...
 $ TMDMOD3   : num  280 285 287 286 287 281 286 288 28
 $ DEMENV5   : num  2207 1243 1492 1809 1731 ...
 $ B07CHE3   : num  37.8 42.1 41.5 39.8 40.4 ...
 $ HIST      : num  1 1 1 1.02 1.29 ...
 $ B13CHE3   : num  125 99.8 133.8 136.6 134.4 ...
 $ B14CHE3   : num  60.3 42.4 47 47.9 43.3 ...
 $ REDL00    : num  19 22 29 53 56 21 52 22 12 17 ...
 $ TWIMRG5   : num  62 81 72 67 65 60 59 61 65 60 ...
 $ LandCover: Factor w/ 13 levels "11","14","20",..:
 $ soilmap   : Factor w/ 19 levels "1","2","3","4",..:
```

```
# save the final table (regression matrix)

write.csv(dat, "02-Outputs/SOC_RegMatrix.csv", row.names = FALSE)
```

Linear regression

- Simple **linear regression** is a statistical method that allows us to summarize and study **relationships** between two continuous (quantitative) variables:
    - variable **X**, is regarded as the **predictor**, explanatory, or independent variable.
    - variable **Y**, is regarded as the response, outcome, or **dependent variable**.

The goal is to build a **mathematical formula** that defines **Y** as a function of the **X** variable.

Once, we built a statistically significant model, it's possible to use it for **predicting** future outcome on the basis of new X values.

Linear regression

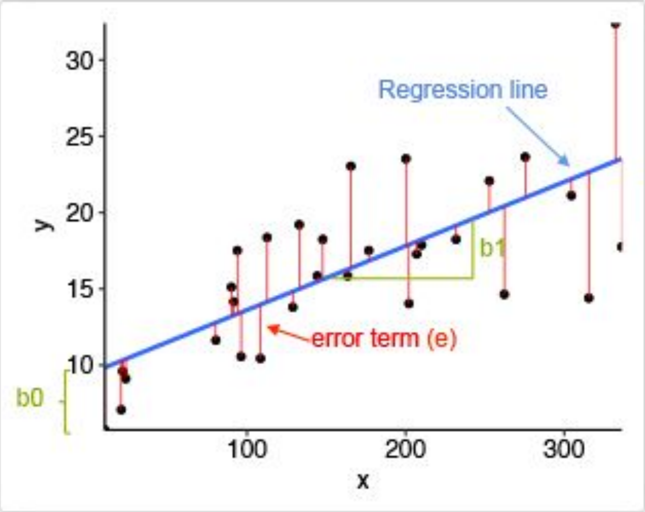
The mathematical formula of the linear regression can be written as follow:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

Y Dependent Variable βn Coefficients X_n Predictors ε Residuals

Graphical representation:

- the best-fit **regression line** is in blue
- the **intercept** (b0) and the **slope** (b1) are shown in green
- the **residuals (errors)** - **e** are represented by vertical red lines



Multiple linear regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

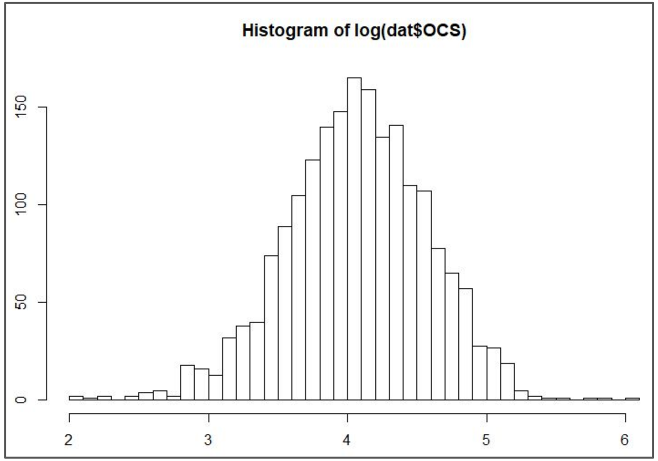
Y Dependent Variable, βn Coefficients, X_n Predictors, ε Residuals

- **Dependent variable Y**: which is to be predicted from a given set of predictors (e.g. organic carbon stocks).
- **Independent variables X's (Predictors):** which influence or explain the dependent variable (covariates)
- **Coefficients β**: values, computed by the **multiple regression**, reflect the relationship and strength of each independent variable to the dependent variable
- **Residuals ε**: The portion of the dependent variable that **cannot be explained by the model**; the model under/over predictions.

Assumptions of linear regression

Linear regression models with standard estimation techniques make a number of **assumptions** about the predictor variables, the response variables, and their relationship. We must **review the assumptions** made when using the model.

Before proceeding with the regression analysis, it is **advisable** to inspect the histogram of the dependent/target variable, in order to see if it has **normal distribution** or it needs to be **transformed** before fitting the regression model.

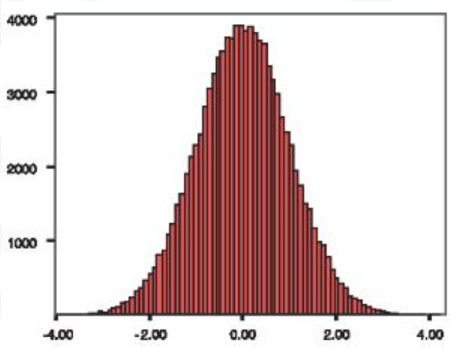


Our data is already **log-transformed**

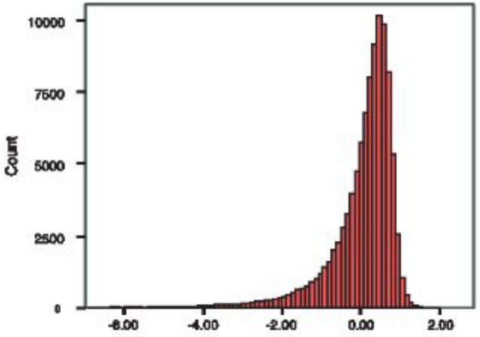
Assumptions of linear regression

- **Normality Assumption:** It is assumed in multiple regression that the **residuals** (prediction errors) are distributed **normally**.

Normal



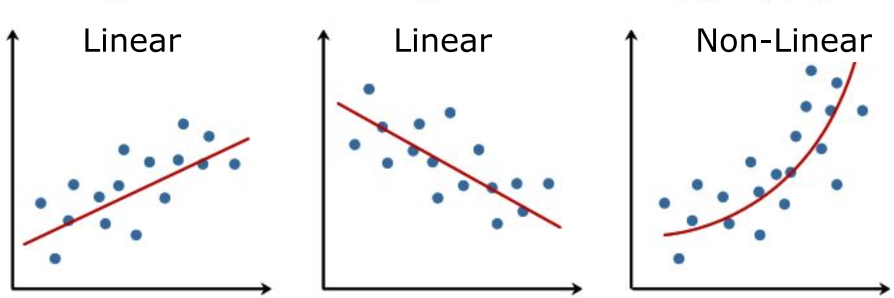
Not normal



- You can produce **histograms** of the residuals **or Q-Q plots**, in order to inspect the distribution of the residual values.

Assumptions of linear regression

- **Linearity Assumption:** The mean value of Y for each specific combination of the X's is a **linear function** of the X's.
- In practice this assumption can virtually never be confirmed, because most relationships in nature are **non-linear**.



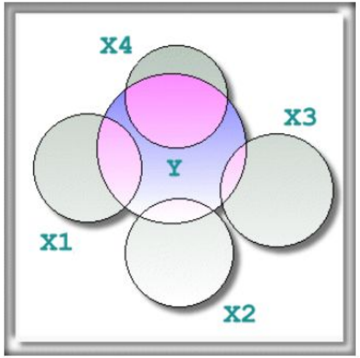
- Fortunately, multiple regression procedures are not greatly affected by minor deviations from this assumption. If curvature in the relationships is evident, you may consider **transforming** the variables.

Assumptions of linear regression

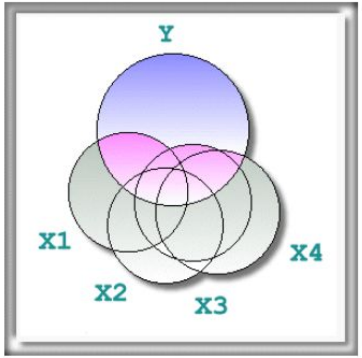
- **Homoscedasticity Assumption:** The variance of error terms is constant for all combinations of X's. The term *homoscedasticity* means *same scatter*.
- A **scatter plot** of standardized residuals versus predicted values can show whether points are equally distributed across all values of the independent variables.

Assumptions of linear regression

- **Collinearity Assumption**: It is assumed that independent variables (Xs) are **not highly correlated** with each other.



No collinearity



Substantial collinearity

- This assumption is tested using Variance Inflation Factor (VIF) values.

Multiple Linear Regression in R

```
# prepare the table for regression (only OCSlog and the covariates)

datdf <- dat[, c("OCSlog", names(covs))]

# Fit a multiple linear regression model

model.MLR <- lm(OCSlog ~ ., data = datdf)

summary(model.MLR)
```

```
Call:
lm(formula = OCSlog ~ ., data = datdf)

Residuals:
     Min       1Q   Median       3Q      Max
-2.17158 -0.25819  0.02733  0.27983  1.82590

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.965e+00  3.190e+00   2.183  0.02918 *
B04CHE3     -5.366e-03  1.294e-03  -4.146 3.58e-05 ***
PRSCHE3      7.099e-04  7.866e-04   0.903  0.36694
TMDMOD3     -9.957e-03  1.140e-02  -0.873  0.38266
DEMENV5     -4.245e-05  8.045e-05  -0.528  0.59785
B07CHE3      9.239e-02  4.631e-02   1.995  0.04622 *
HIST         1.124e-01  2.802e-02   4.011 6.37e-05 ***
```

Multiple Linear Regression in R

- Note that factor variables were turned into dummies:

```
LandCover130 -8.341e-02  7.413e-02  -1.125  0.26067
LandCover150 -1.246e-01  7.573e-02  -1.645  0.10013
LandCover190 -2.664e-02  1.142e-01  -0.233  0.81565
LandCover210  2.694e-01  3.144e-01   0.857  0.39160
soilmap2      7.682e-02  1.053e-01   0.729  0.46599
soilmap3      2.405e-01  1.412e-01   1.703  0.08875 .
soilmap4      1.594e-01  2.404e-01   0.663  0.50745
soilmap5      6.411e-01  1.149e-01   5.582 2.85e-08 ***
soilmap6      5.574e-01  1.291e-01   4.317 1.69e-05 ***
```

- Note significance codes and model statistics:

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

Residual standard error: 0.4309 on 1414 degrees of freedom
Multiple R-squared:  0.2891,    Adjusted R-squared:  0.269
F-statistic: 14.38 on 40 and 1414 DF,  p-value: < 2.2e-16
```

Characteristics of the model

R-squared (**R^2**) is the **percentage of variation** in the dependent variable that is **explained** by the model. It ranges between 0 and 1 (0% to 100%).

- The higher the R^2 value is, the better the model **fits** your data.
- R^2 increases when **additional predictors** are added in the model.
- **Adjusted R^2** increases **only** when the new variable actually has a **significant effect** on the predicted value.

P-value is the measure of **statistical significance** of the model

- To determine whether the model is statistically significant, compare the p-value to your **significance level** (usually, a significance level of **0.05** works well)
- **p-value ≤ significance level:** The relationship is statistically significant. You may proceed with modelling.
- **p-value > significance level:** The relationship is not statistically significant. You may need a new model.

Characteristics of the model

Questions:

- What is the $\mathbf{R^2}$ of the model?
- Is it high or low?
- What is the **p-value** of the model?
- Is the model **statistically significant** or not?
- Are all the **predictors** significant?

Stepwise variable selection

- There were many **insignificant predictors** in the model. Let's try to optimise the model using stepwise procedure
- **Stepwise regression** can be achieved either:
  - by trying out **one independent variable at a time** and including it in the regression model if it is statistically significant,
  - or by including all potential independent variables in the model and **eliminating** those that are **not statistically significant**,
  - or by a combination of **both methods**.
- At each step, the significance is tested, using F-tests, t-tests, adjusted R squared or other methods;
- The **goal** is to find a **set of independent variables** which **significantly influence** the dependent variable in the linear model.

Stepwise MLR

```
# stepwise variable selection

model.MLR.step <- step(model.MLR, direction="both")
```

```
Step:  AIC=-2426.97
OCSlog ~ B04CHE3 + PRSCHE3 + B07CHE3 + HIST + B13CHE3 + REDL00 +
    soilmap

            Df Sum of Sq    RSS     AIC
<none>                   265.17 -2427.0
+ TMDMOD3    1    0.1988 264.97 -2426.1
+ TWIMRG5    1    0.1113 265.06 -2425.6
+ B14CHE3    1    0.0685 265.10 -2425.3
+ DEMENV5    1    0.0013 265.17 -2425.0
- B07CHE3    1    0.9421 266.11 -2423.8
- B13CHE3    1    1.1703 266.34 -2422.6
- PRSCHE3    1    1.3411 266.51 -2421.6
+ LandCover 12    2.2398 262.93 -2415.3
- B04CHE3    1    3.8768 269.05 -2407.8
- HIST       1    4.0319 269.20 -2407.0
- REDL00     1    6.4985 271.67 -2393.7
- soilmap   18   29.4588 294.63 -2309.7
```

Stepwise MLR

```
# summary of the new model using stepwise covariates selection

summary(model.MLR.step)
```

```
Residual standard error: 0.4306 on 1430 degrees of freedom
Multiple R-squared:  0.2821,	Adjusted R-squared:   0.27
F-statistic: 23.41 on 24 and 1430 DF,  p-value: < 2.2e-16
```

Questions:

- Is the $\mathbf{R^2}$ higher or lower than before stepwise selection?
- Is the adjusted $\mathbf{R^2}$ higher or lower than before?
- Is the model statistically significant?

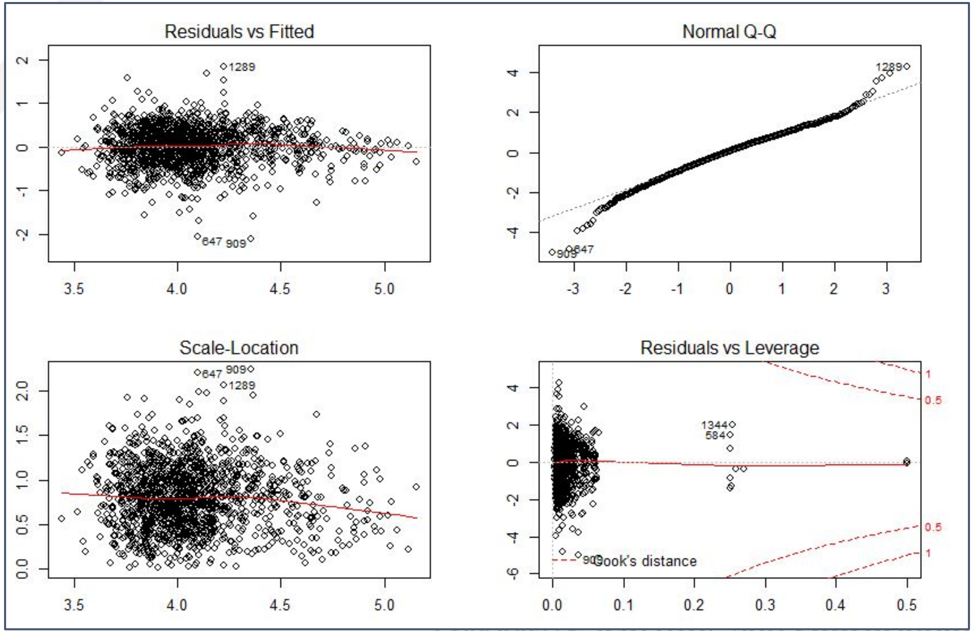
Stepwise MLR

```
# analysis of variance (anova) of the new model

anova(model.MLR.step)
```

```
Analysis of Variance Table

Response: OCSlog
            Df  Sum Sq Mean Sq  F value    Pr(>F)    
B04CHE3      1  50.978  50.978 274.9128 < 2.2e-16 ***
PRSCHE3      1   5.444   5.444  29.3554 7.061e-08 ***
B07CHE3      1   2.620   2.620  14.1303 0.0001774 ***
HIST         1   8.920   8.920  48.1031 6.106e-12 ***
B13CHE3      1   2.282   2.282  12.3060 0.0004655 ***
REDL00       1   4.482   4.482  24.1679 9.848e-07 ***
soilmap     18  29.459   1.637   8.8257 < 2.2e-16 ***
Residuals 1430 265.172   0.185                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

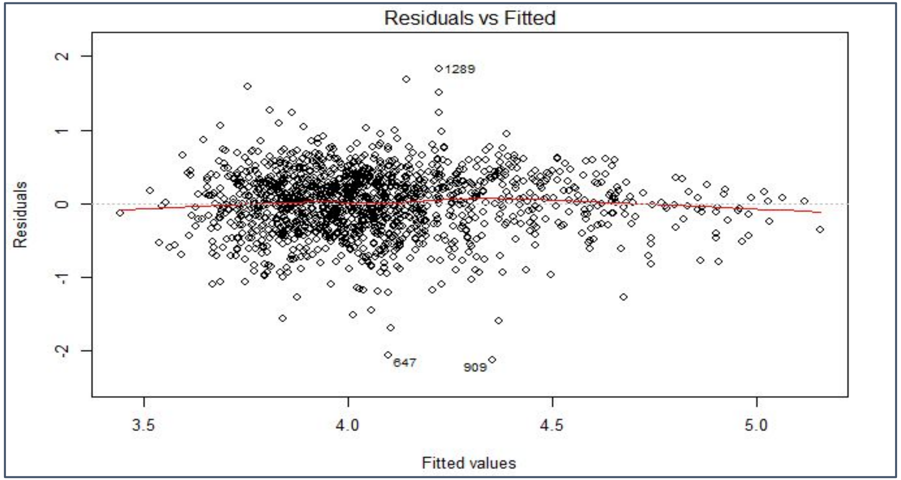
Graphical diagnosis of MLR

```
# graphical diagnosis of the regression analysis
par(mfrow=c(2,2))
plot(model.MLR.step)
par(mfrow=c(1,1))
```



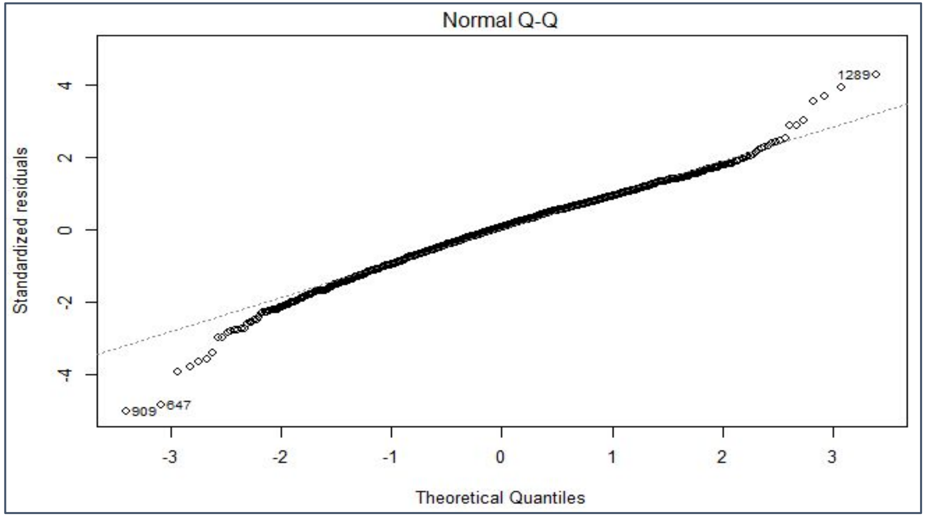
Residuals vs Fitted

- This plot shows if residuals have **non-linear patterns**.
- If you find equally spread residuals around a horizontal line without distinct patterns, that is a good indication you don't have non-linear relationships.



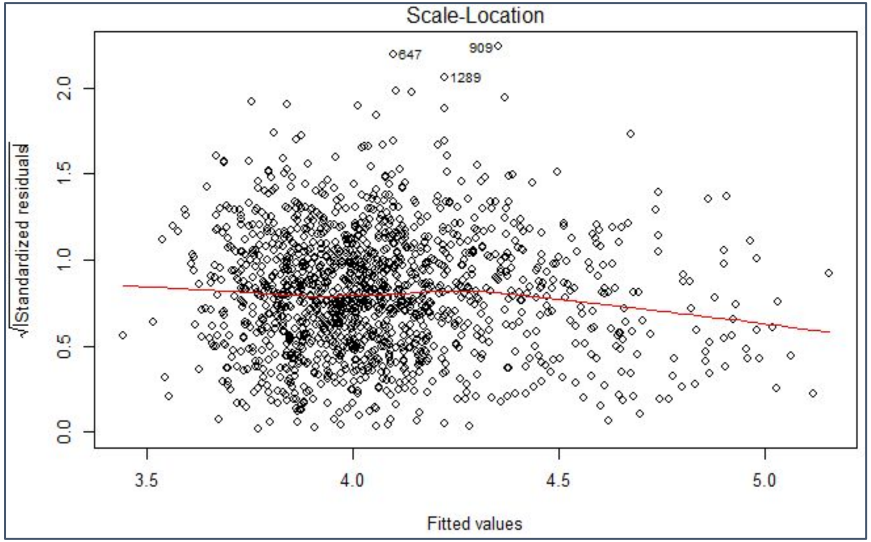
Normal Q-Q plot

- This plot shows if residuals are normally distributed (**checking normality assumption**)
- It's good if residuals are lined well on the straight dashed line.



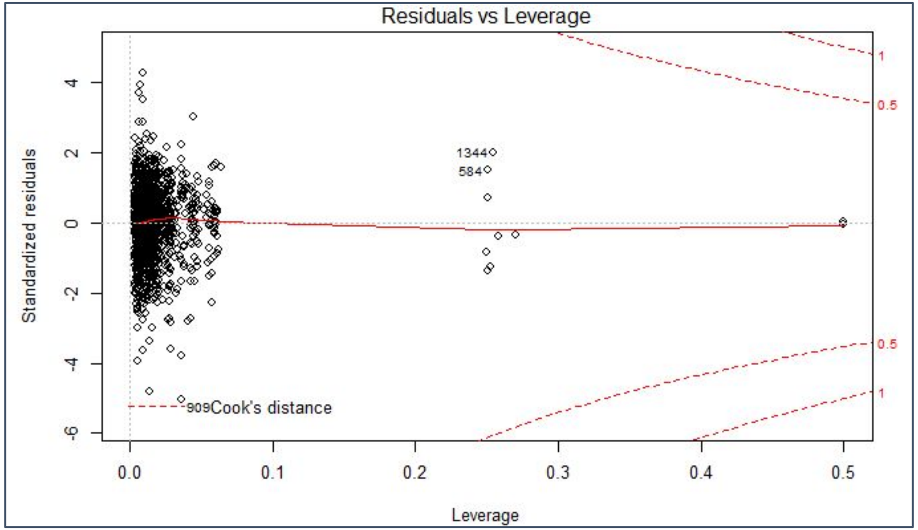
Scale-Location plot

- This plot shows if residuals are spread equally along the ranges of predictors (**checking homoscedasticity assumption**).
- It's good if you see a horizontal line with equally (randomly) spread points.



Residuals vs Leverage

- This plot helps us to find influential cases (**outliers that greatly affect the model**) if any.
- We watch out for outlying values at the upper right corner or at the lower right corner.



Collinearity test, Bonferroni test

```
# collinearity test using variance inflation factors

library(car)

vif(model.MLR.step)

# problematic covariates should have

# sqrt(VIF) > 2

sqrt(vif(model.MLR.step))

# Removing a layer from the stepwise model

# model.MLR.step <- update(model.MLR.step, . ~ . - PRSCHE3)

# outlier test using the Bonferroni test

outlierTest(model.MLR.step)
```

```
> sqrt(vif(model.MLR.step))
            GVIF       Df GVIF^(1/(2*Df))
B04CHE3 3.586117 1.000000        1.893705
PRSCHE3 3.862819 1.000000        1.965406
B07CHE3 3.814856 1.000000        1.953166
HIST    1.422848 1.000000        1.192832
B13CHE3 3.406162 1.000000        1.845579
REDL00  1.347996 1.000000        1.161032
soilmap 1.788046 4.242641        1.016273
```

```
> outlierTest(model.MLR.step)
      rstudent unadjusted p-value Bonferroni p
909  -5.067332         4.5609e-07   0.00066406
647  -4.869921         1.2405e-06   0.00180610
1289  4.294399         1.8698e-05   0.02722400
```

Mapping OCS using the MLR model

- After thoroughly checking the model, we can use it for **prediction;**
- We predict the **OCSlog** values for all unknown locations using our **model and covariates**;

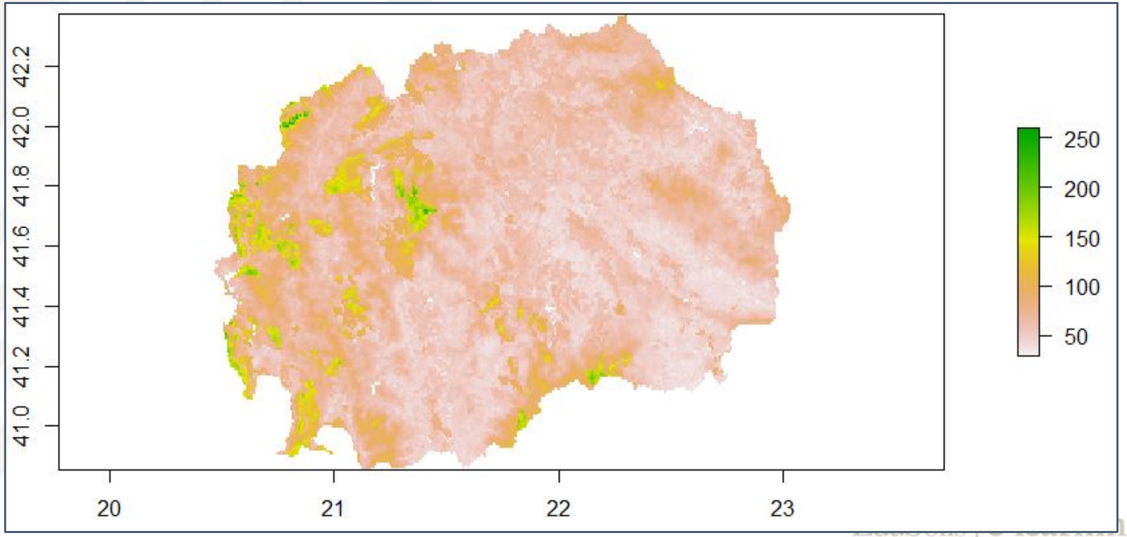
```
# Make a prediction across all Macedonia using linear model

pred <- predict(covs, model.MLR.step)
```

- Object ‘pred’ is a raster with predicted OCSlog across all the the country
- Now, we need to back-transform it from log to OCS (t/ha)

```
# Back transform predictions log transformed

pred <- exp(pred)
```

Explore and save the MLR map

```
# Explore and save the result as a tiff file
plot(pred)
writeRaster(pred, filename = "02-Outputs/MKD_OCS_MLR.tif",
                overwrite=TRUE)
```



Regression step finished

- We created our **first map** using predictions of multiple linear regression!
- However, the model is not perfect: it still contains **residuals** (difference between predicted and observed value),
- If the residuals are **autocorrelated** (have a spatial pattern), then the map can be further improved by adding a kriging step.

Next step - kriging!