Food and Agriculture Organization of the United Nations

AFACI Asian Food & Agriculture Cooperation Initiative

Rural Development Administration

EduSoils

Uncertainty and Validation

in digital soil mapping

GLOBAL SOIL PARTNERSHIP

International Training on Digital Soil Property Mapping and Information Delivery, 15-19 November 2021, Yi Peng & Isabel Luotto

What is uncertainty?

- **Soil mapping** involves making predictions at locations where n**o soil measurements** were taken.
- This inevitably leads to **prediction errors** because soil spatial variation is complex and **cannot be modeled perfectly**.
- In fact, we may even be uncertain about the soil at the measurement locations because **no measurement method is perfect** and uncertainty also arises from measurement errors.
- **Uncertainty is an acknowledgement of error**: we are aware that our representation of reality may differ from reality and express this by being uncertain

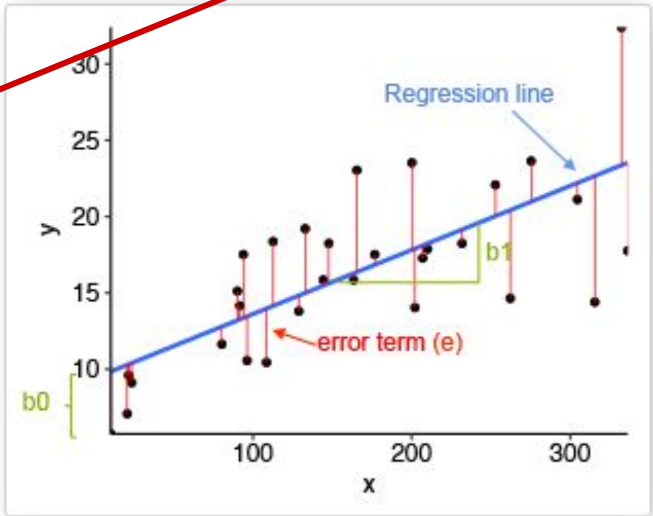
Sources of uncertainty

- Attribute uncertainty of soil measurements
- Positional uncertainty of soil measurements
- Uncertainty in covariates
- Uncertainty in prediction models

Uncertainty of the linear model

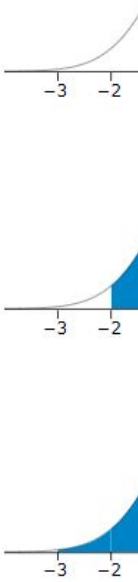
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

Y Dependent Variable βn Coefficients X_n Predictors ε Residuals



Uncertainty of the linear model

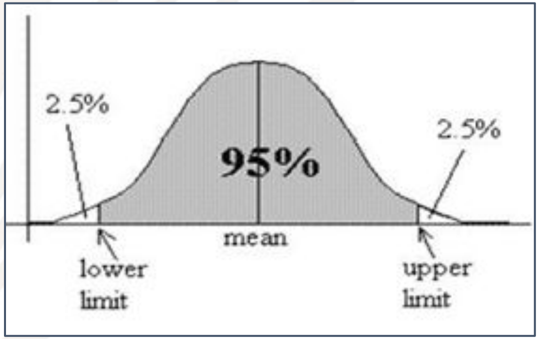
- In the presence of uncertainty, we **cannot identify** a single, **true values** for each pixel of the map.
- But we can identify **all possible values** and a probability for each one - to characterise the uncertain variable with a **probability distribution.**
- If the distribution is **normal**, it is easy to construct a **confidence interval**, where e.g. we are certain with 95% confidence that the true value will be within **2 standard deviations** from the mean (prediction)

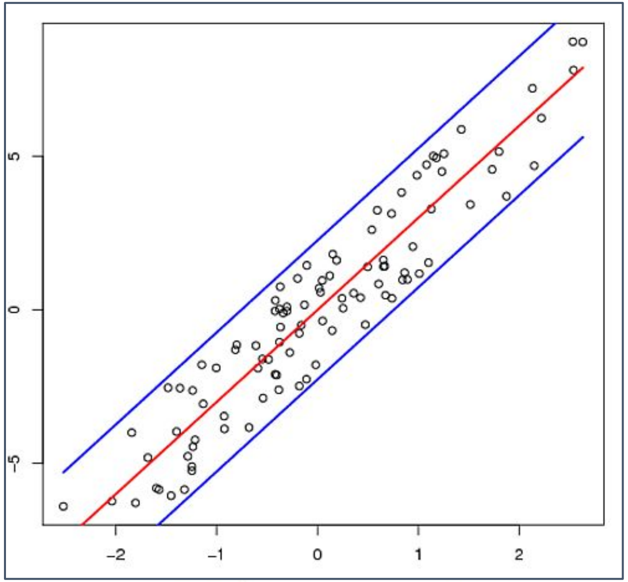


Confidence interval

95% confidence interval:

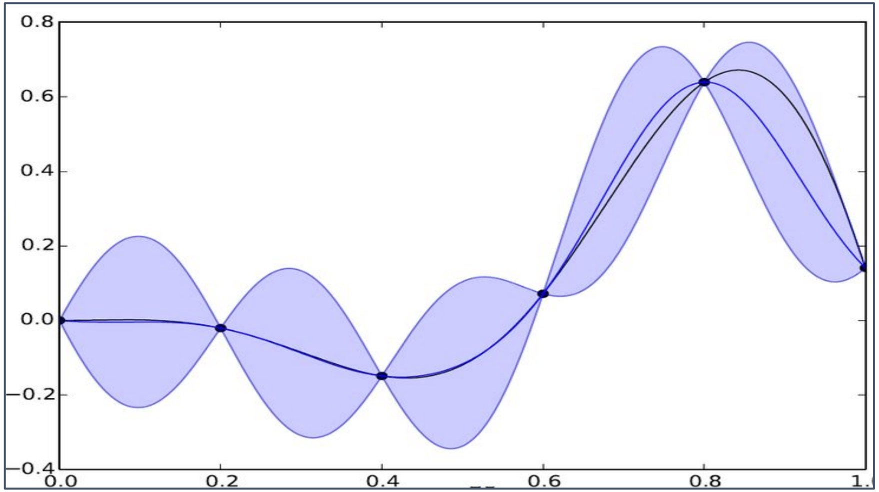
We are certain that the unknown value lies within **±2sd** from the **predicted value.**





Uncertainty of Kriging

- Kriging **reduces uncertainty** around the sampling points where we have **observations**;
- The **more distance** from the the sampling points - **the higher uncertainty.**

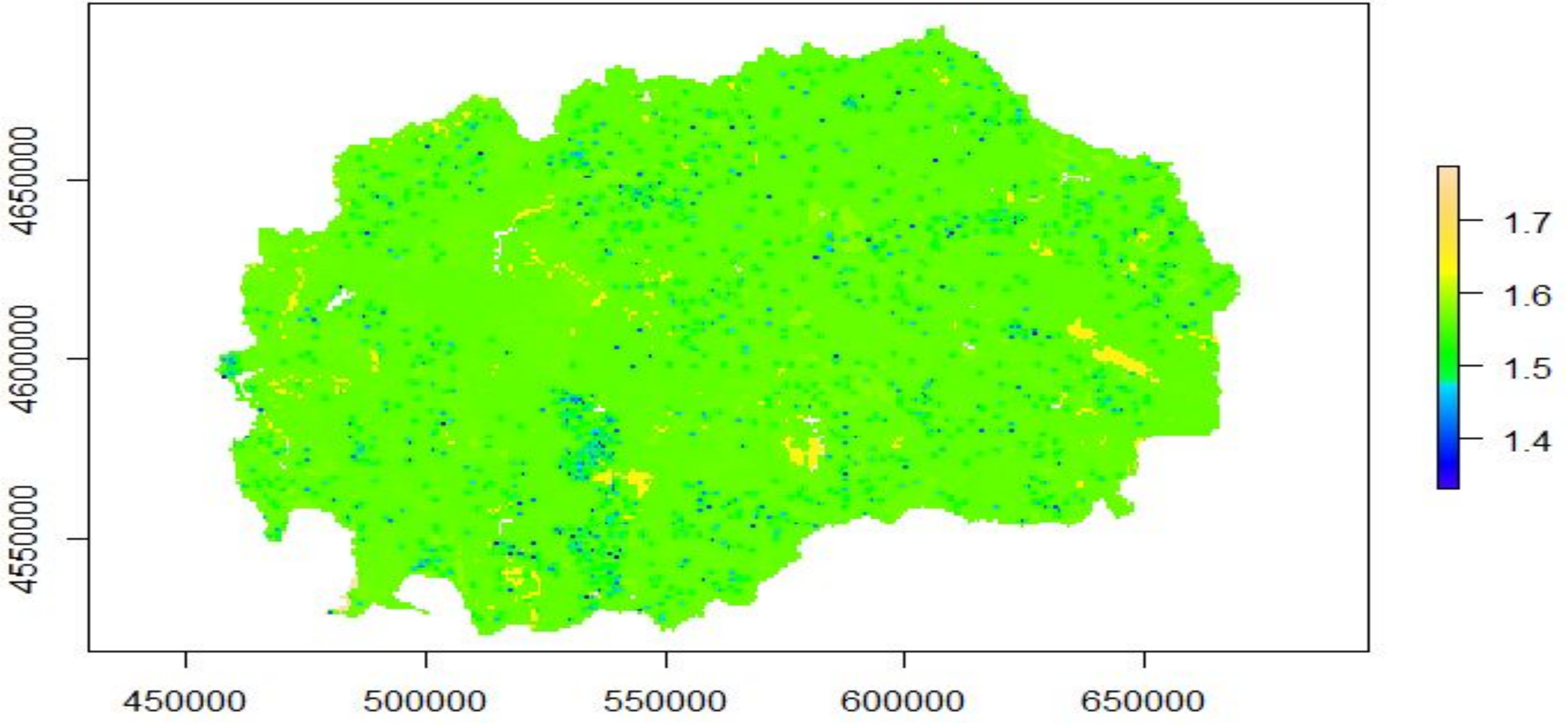


Uncertainty of Regression Kriging

- For **Regression Kriging**, uncertainty of the model includes both **Linear** component and **Kriging** component.
- We can use **standard deviation** as a **quantification of uncertainty** in every point of the map, and build confidence intervals.
- For Regression Kriging, standard deviation can be easily derived as a **square root of kriging variance** (we did it yesterday):

```
# Make an uncertainty estimation as a map of standard deviations
# Standard deviation is the square root of kriging variance
RKsd <- exp(sqrt(raster(pred_gstat, layer='var1.var')))
plot(RKsd2, col=  topo.colors(255))
writeRaster(RKsd,'02-Outputs/MKD_OCS_RK_sd.tif', overwrite=TRUE)
```

Map of standard deviations



Uncertainty of Random Forest

- **Machine learning** models typically yield **more accurate predictions** but quantification of the associated **uncertainty** is more **difficult.**

- Random Forest is a **non-linear** model, the residuals may be **not normally distributed**, therefore we **cannot** quantify uncertainty in the same way as for Regression Kriging.

Uncertainty of Random Forest

- The **research** for the best way to quantify uncertainty of Random Forest is **ongoing**.
- The most promising approach makes use of **quantile regression forests** (Meinshausen, 2006; Vaysse and Lagacherie, 2017).
- The **example R code** to realize this approach is included in the SOC mapping **Cookbook** (FAO, 2018), and was also sent to your **email**.
- Be aware that the **quantile regression forests** algorithm is very **<u>computationally intensive</u>,** and requires a lot of processing time and computer memory.

What is Validation?

<u>No map is perfect</u>

- All maps, including soil maps, are **representations of reality** that are often based on an underlying model.
- There is always a **deviation** between the phenomenon **depicted on the map** and the phenomenon **observed in the real world**, i.e. each map contains **errors**.
- The magnitude of the errors determines the **quality of the map**.
- If a map matches reality well (the error is small), the quality or accuracy of the map is high. On the other hand, if a map does not match reality well, map accuracy is low.

What is Validation?

- It is important that the **quality of a map** is determined and **quantified** through (statistical) **validation**.
- **Validation** is defined here as an activity in which the soil map **predictions** are **compared** with **observed** values. From this comparison, the map quality can be quantified and summarized using **map quality measures.**
- **Quality measures** obtained through validation are **global** measures: each quality measure gives **one value for the entire map**.
- Note that this is different from results obtained through **uncertainty** assessment which is quantified **for each pixel** of the map.

Validation methods

To perform **validation**, we need to compare our map's predictions with observed values which **were not used for calibration of the model**. This can be achieved in one of the 3 ways:

1. **Additional probability sampling**: a new soil survey to collect data from the field and compare it to the predicted values;

   Advantages:

- Allows to make **unbiased** quantification of map quality

   Disadvantages:

- **High cost** of soil survey

Validation methods: data splitting

2. **Data splitting**: use part of the data (e.g. 75%) for calibrating the model, and other part (e.g. 25%) for validating it.

Advantages:

- No need for field survey

Disadvantages:

- If data is **sparse** then difficult to split it into representative parts, and the result may be **biased**

Validation methods: data splitting

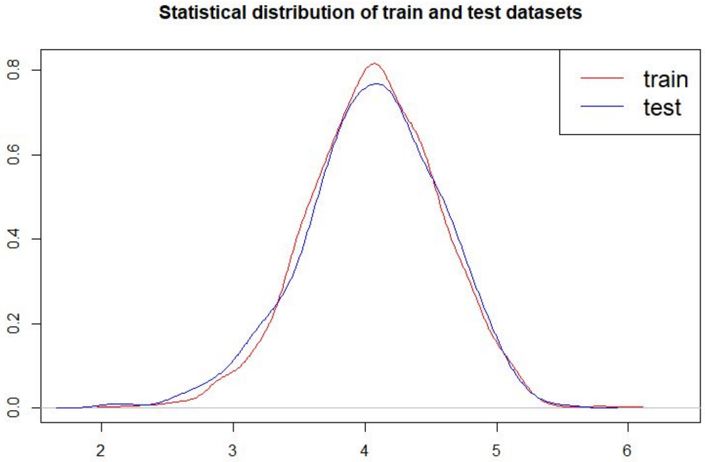
- During the **data preparation** we already split our initial dataset into training data - for **calibration** of the model, and test data - for **validation:**

```r
library(caret)
# Define the random numbers table (to get reproducible result)
set.seed(11042019)

# Create random selection of 75% of the data as 'train' dataset and
# 25% as 'test' dataset
train.ind <- createDataPartition(1:nrow(dat), p = .75, list = FALSE)
train <- dat[ train.ind,]
test  <- dat[-train.ind,]
```

Validation methods: data splitting

- When splitting the dataset it is important to ensure that the **distribution is same** both for calibration (training) data, and for validation (testing) data, in order to **minimize bias**.

```
> summary(train$OCS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  7.643  42.214  59.077  66.307  81.880 425.379
> summary(test$OCS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  7.801  42.107  58.986  65.874  83.201 249.767
```



Validation methods: cross-validation

3. **K-fold Cross-validation:** the dataset is split into K (e.g. 10) roughly equal sets (folds), then for each set a model is calibrated. Validation results are then summarised for all folds.

Advantages:

- Uses all data for both calibration and validation - better than splitting **when the data is limited**

Disadvantages:

- Like data splitting, it may be **biased**

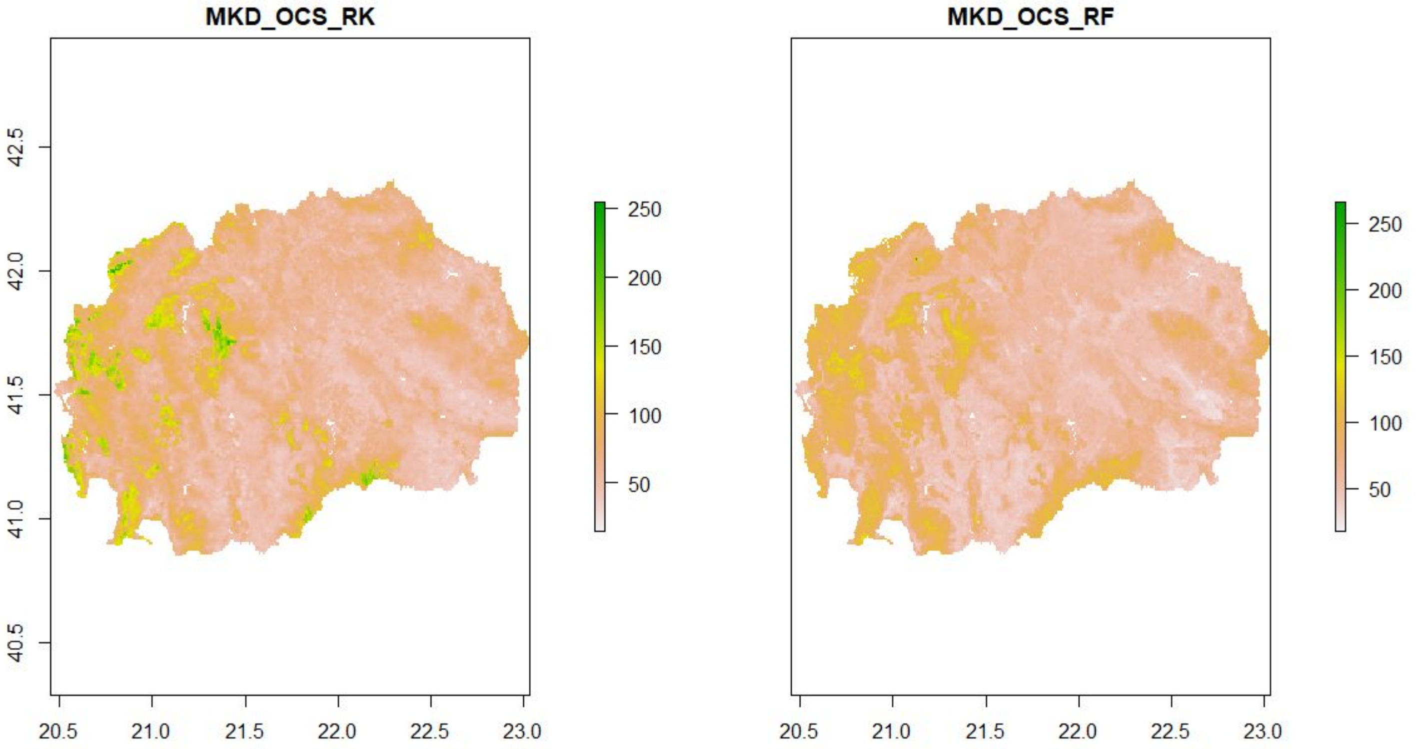
els.

*Example R codes for cross-validation using **caret** package are available in the SOC mapping **Cookbook** (FAO, 2018)*

Validation in R

- Let's create a script for validation in R

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

# Load and stack the maps from the results folder
RKmap<-raster("02-Outputs/Final Maps/MKD_OCS_RK.tif")
RFmap<-raster("02-Outputs/Final Maps/MKD_OCS_RF.tif")
maps <- stack(RKmap, RFmap)

# Explore the maps
names(maps)
summary(maps)
plot(maps)
```


MKD_OCS_RK
MKD_OCS_RF
42.5
42.0
41.5
41.0
40.5
20.5
21.0
21.5
22.0
22.5
23.0
250
200
150
100
50

Extracting predictions to points

```
# Load the validation dataset.
# It was was prepared in the 'data_preparation_profiles' script
test <- read.csv("02-Outputs/dat_test.csv")
# Promote to spatialPointsDataFrame and set crs
coordinates(test) <- ~ X + Y
test@proj4string <- CRS(projargs = "+init=epsg:4326")
# Extract the predicted values from the maps to the validation dataset
test <- extract(x = maps, y = test, sp = TRUE)
summary(test)
```

```
      OCS                OCSlog          MKD_OCS_RK        MKD_OCS_RF
 Min.   :  7.801   Min.   :2.054   Min.   : 24.21   Min.   : 32.43
 1st Qu.: 42.107   1st Qu.:3.740   1st Qu.: 46.38   1st Qu.: 47.55
 Median : 58.986   Median :4.077   Median : 55.62   Median : 54.99
 Mean   : 65.874   Mean   :4.057   Mean   : 60.84   Mean   : 60.82
 3rd Qu.: 83.201   3rd Qu.:4.421   3rd Qu.: 65.97   3rd Qu.: 67.46
 Max.   :249.767   Max.   :5.521   Max.   :199.26   Max.   :265.51
                                   NA's   :3        NA's   :3
```

Prediction errors

```
# Remove NA values

test<-as.data.frame(test)

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

- **Prediction Errors** (PE) are a difference between **predicted** (on the map) and **observed** (true) values:

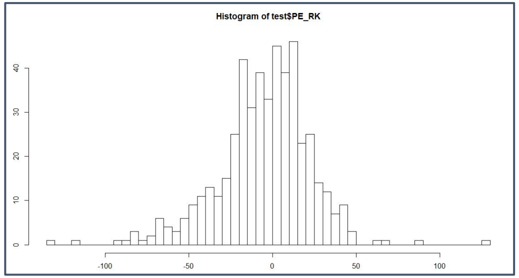
PE = Predicted - Observed

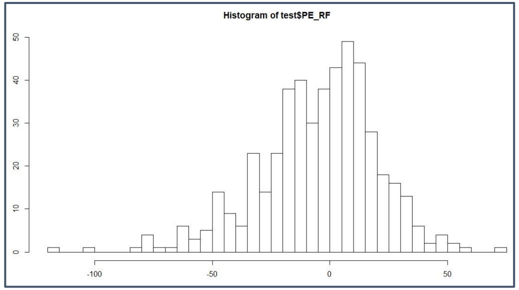
```
# Calculate prediction errors

test$PE_RK <- test$MKD_OCS_RK - test$OCS

test$PE_RF <- test$MKD_OCS_RF - test$OCS
```

Prediction errors

```
# Explore prediction errors

summary(test$PE_RK)

summary(test$PE_RF)



hist(test$PE_RK, breaks=50)

hist(test$PE_RF, breaks=50)
```

```
> summary(test$PE_RK)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-133.300  -18.700   -1.711   -4.901   12.600  127.600
> summary(test$PE_RF)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-119.500  -18.020   -2.475   -4.914   11.420   74.930
```





Map quality measures

We will use the following map quality measures:

- Mean error (ME)
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Amount of Variance Explained (AVE)

Mean error (ME) is defined as the population mean (spatial mean) of the prediction errors. **ME** measures **bias** in the predictions.

ME should be (close to) **zero**, which means that predictions are unbiased meaning that there is no systematic **over-** or **under-prediction** of the soil property of interest.

Map quality measures

Mean absolute error (MAE) and **mean squared error** (MSE) are measures of map accuracy and indicate the **magnitude of error** we make on average. The **MAE** is defined by the population mean of the **absolute** errors, and the **MSE** by the population mean of the **squared** errors.

Many authors report the **root mean squared error** (RMSE) instead of the MSE, which is computed by taking the **square root of the MSE**. The **RMSE** can be a more appealing quality measure since it has the **same unit** of measurement as the mapped property and, therefore, can be **more easily interpreted**.

Map quality measures

Amount of Variance Explained (AVE) is the measure of **model efficiency. AVE** quantifies the fraction of the variation in the data that is explained by the prediction model.

AVE is similar to **R-squared** (**R^2**) of the model. The maximum value of AVE is 1 (meaning that the model describes 100% of variation)

AVE measures the improvement of the model prediction over using the mean of the data set as predictor. In case the **AVE** is negative (**<0**), then the mean of the data set is a better predictor than the prediction model.

Calculating map quality measures

```
# Regression Kriging


# Mean Error

ME_RK <- mean(test$PE_RK)

# Mean Absolute Error (MAE)

MAE_RK <- mean(abs(test$PE_RK))

# Root Mean Squared Error (RMSE)

RMSE_RK <- sqrt( sum(test$PE_RK^2) / length(test$PE_RK) )

# Amount of Variance Explained (AVE)

AVE_RK <- 1 - sum(test$PE_RK^2) / sum( (test$OCS - mean(test$OCS))^2 )
```

Values	
AVE_RK	0.298673142223776
MAE_RK	20.8234780716041
ME_RK	-4.90080062434658
RMSE_RK	28.2668431744963

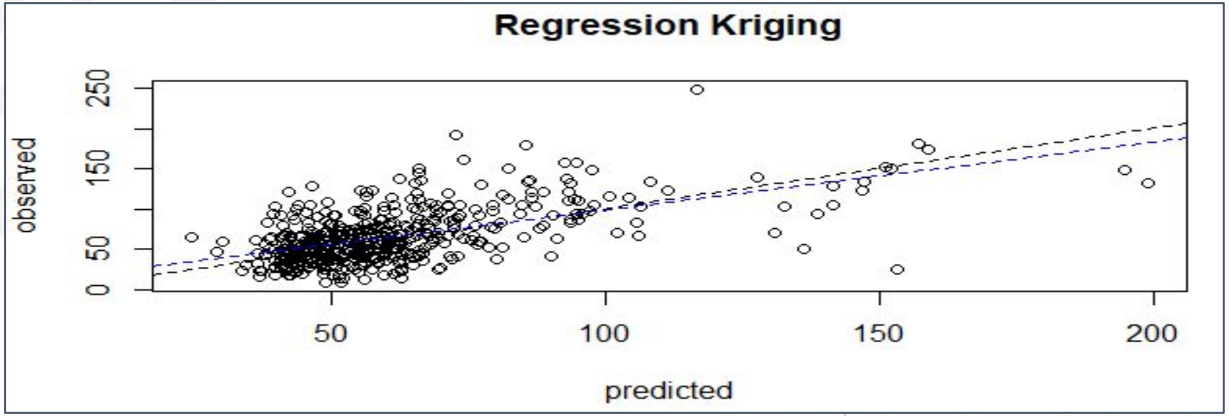
Calculating map quality measures

```
# Random Forest


# Mean Error

ME_RF <- mean(test$PE_RF)

# Mean Absolute Error (MAE)

MAE_RF <- mean(abs(test$PE_RF))

# Root Mean Squared Error (RMSE)

RMSE_RF <- sqrt(sum(test$PE_RF^2) / length(test$PE_RF))

# Amount of Variance Explained (AVE)

AVE_RF <- 1 - sum(test$PE_RF^2) / sum( (test$OCS - mean(test$OCS))^2 )
```

Values	
AVE_RF	0.418701629637483
AVE_RK	0.298673142223776
MAE_RF	19.4869450714773
MAE_RK	20.8234780716041
ME_RF	-4.9135509382642
ME_RK	-4.90080062434658
RMSE_RF	25.7345528595679
RMSE_RK	28.2668431744963

Graphical quality measures

```
# scatter plot
plot(test$MKD_OCS_RK, test$OCS, main="Regression Kriging",
xlab="predicted",ylab='observed')
# 1:1 line in black
abline(0,1, lty=2, col='black')
# regression line between predicted and observed in blue
abline(lm(test$OCS ~ test$MKD_OCS_RK), col = 'blue', lty=2)
```



Graphical quality measures

```
# scatter plot
plot(test$MKD_OCS_RF, test$OCS, main="Random Forest", xlab="predicted",ylab='observed')
# 1:1 line in black
abline(0,1, lty=2, col='black')
# regression line between predicted and observed in blue
abline(lm(test$OCS ~ test$MKD_OCS_RF), col = 'blue', lty=2)
```