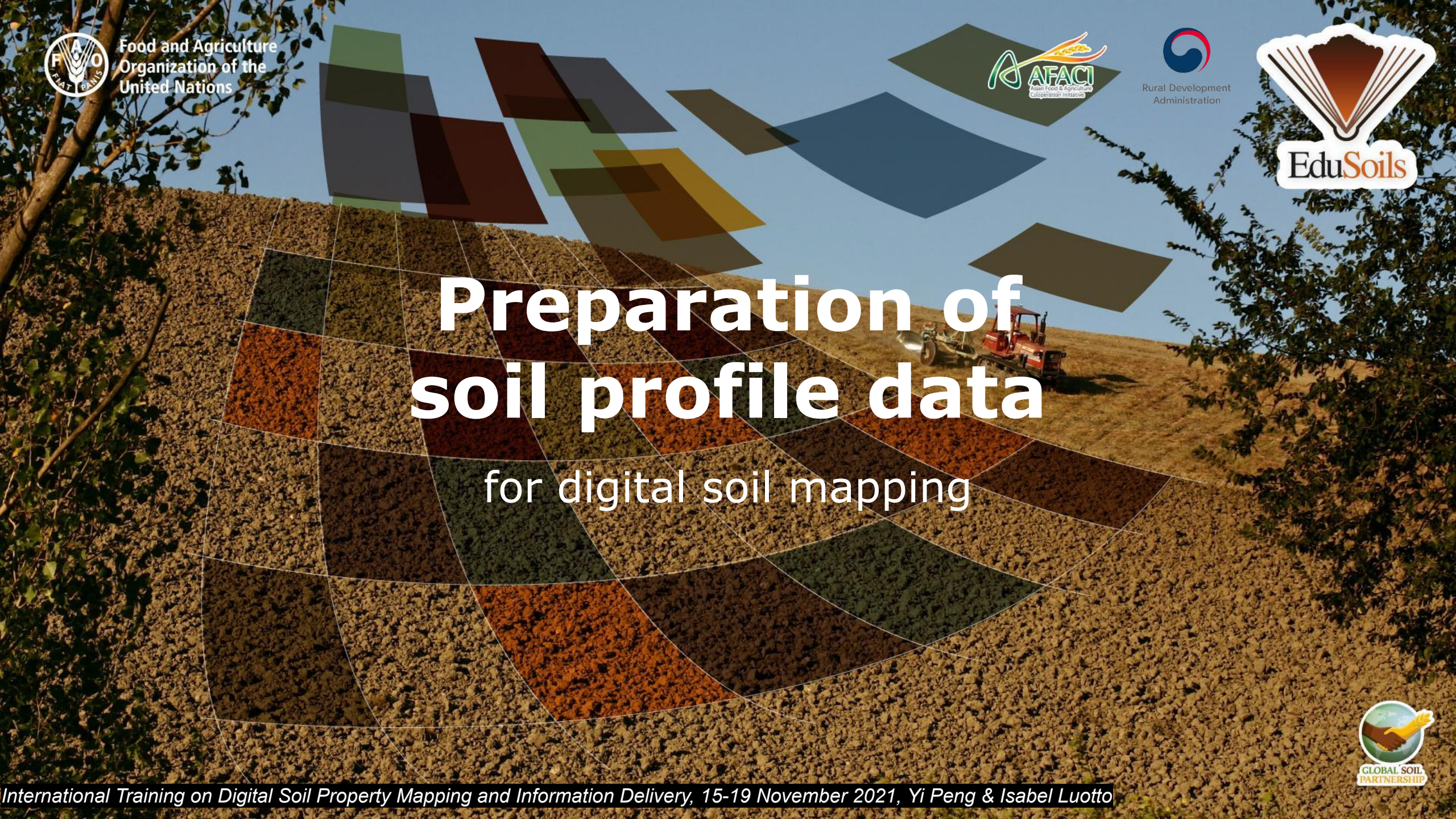

Food and Agriculture Organization of the United Nations
AFACI
Asian Food & Agriculture Cooperation Initiative
Rural Development Administration
EduSoils
Preparation of soil profile data
for digital soil mapping
GLOBAL SOIL PARTNERSHIP

Table of contents:

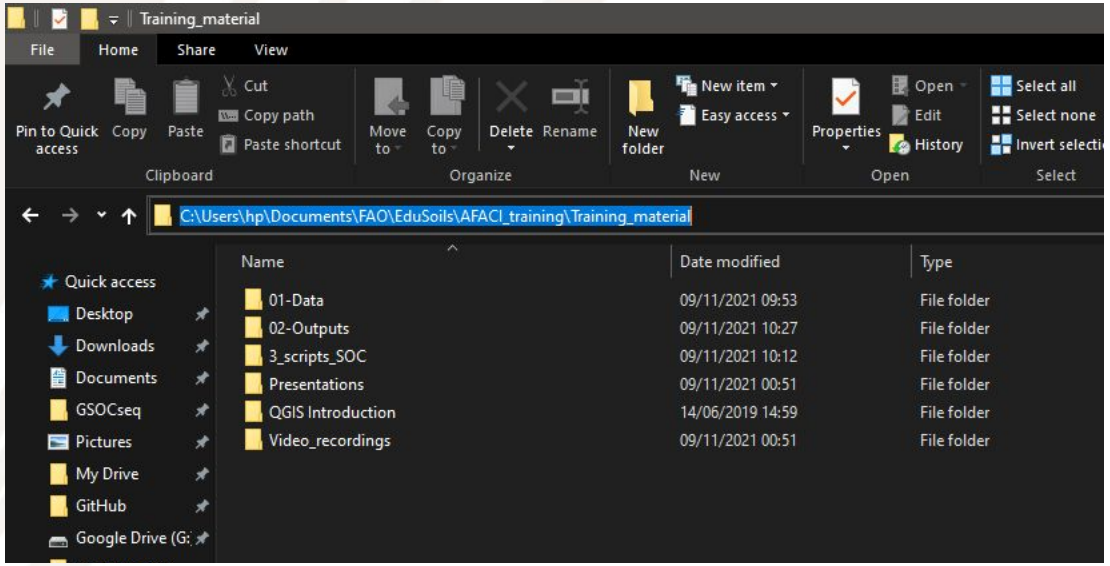
Introduction

- In this course we will use the **training data** from the Macedonian Soil Information System Database (MASIS - FAO 2015).
- The data can be found in the **Training Material** folder
- The course will provide step-by-step guidance to map soil properties, using the example of **SOC stocks**. Same procedure may be followed for mapping other properties.
- For each of the steps we will create an **R script**:

1. Preparation of soil profile data
2. Preparation of environmental covariates
3. Modelling with Regression Kriging
4. Modelling with Random Forest
5. Validation and Uncertainty

Let’s get started!

- Open R-studio and create a new script
- Save it as ‘Data preparation profiles’
- Set the working directory



Let's get started!

- Open R-studio and create a new script
- Save it as 'Data preparation profiles'
- Set the working directory

```
# Set working directory
setwd("C:/Users/hp/Documents/FAO/EduSoils/AFACI_training/Training_material")
```

- don't forget to change the single **** to a double one **** or to **/**

Soil Profile Data

- The first task is to preparing the soil profile data
- Usually, profile data comes in separate 2 tables:
  - A table with profile location and description;
  - A table with all horizons and associated soil properties;
- We will have to **merge** these data into a data set;
- Then, we will need to **explore, check and clean** the data!

The quality of the input data is what defines the quality of the final map!

Import soil layers (horizons)

```
# Import soil layers (horizons) data from a .csv table
dat_layers <- read.csv(file = "01-Data/horizons.csv")
# Explore the data
str(dat_layers)
```

- In the console you can see the 'data frame' structure and all the attributes that are included in the table

```
> str(dat_layers)
'data.frame':   5015 obs. of  16 variables:
 $ ProfID           : Factor w/ 1958 levels "P0004","P0007",..: 372 372 372 373 3
 $ HorID            : Factor w/ 4998 levels "P0004H01","P0004H02",..: 783 784 785
 $ DepthFrom        : int  3 30 62 0 18 43 80 0 0 20 ...
 $ DepthTo          : int  30 60 90 18 43 85 110 30 20 40 ...
 $ CaCO3_perc       : num  11.9 2.6 11.6 1.9 1.9 5.6 4 0 0 0 ...
 $ OM_perc          : num  2.4 2.1 1.8 3 2.5 1.5 1.6 3.6 2.1 1.7 ...
 $ pH_H2O           : num  8.2 8.4 8.6 8.2 8 8.2 8 6.2 6 6.9 ...
 $ pH_nKCl          : num  7.4 7.3 7.6 7.3 7 7.2 7.6 5.2 NA NA ...
 $ P2O5_mg100g      : num  1 1 1 19.2 6.2 1 NA 11.7 NA NA ...
 $ K2O_mg100g       : num  25 25 17.8 25 15 7.3 NA 16.6 NA NA ...
 $ Bulk_density_gcm3: num  NA NA NA NA NA NA NA NA NA NA ...
 $ Higroscopic_water: num  NA NA NA NA NA NA NA NA NA NA ...
 $ Stones_perc      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sand_perc        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Silt_perc        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Clay_perc        : num  0 0 0 0 0 0 0 0 NA NA ...
>
```

Explore soil layers (horizons)

- To open the imported table you may click on the **dat_layers** in top-right 'Data' window or type:

```
View(dat_layers)
```

Filter

	ProfID	HorID	DepthFrom	DepthTo	CaCO3_perc	OM_perc	pH_H2O	pH_nKCl	P2O5_mg100g	K2O_mg100g	Bulk_density_gcm3	Higr
1	P0857	P0857H01	3	30	11.9	2.4	8.2	7.4	1.0	25.0	NA	
2	P0857	P0857H02	30	60	2.6	2.1	8.4	7.3	1.0	25.0	NA	
3	P0857	P0857H03	62	90	11.6	1.8	8.6	7.6	1.0	17.8	NA	
4	P0863	P0863H01	0	18	1.9	3.0	8.2	7.3	19.2	25.0	NA	
5	P0863	P0863H02	18	43	1.9	2.5	8.0	7.0	6.2	15.0	NA	
6	P0863	P0863H03	43	85	5.6	1.5	8.2	7.2	1.0	7.3	NA	
7	P0957	P0957H04	80	110	4.0	1.6	8.0	7.6	NA	NA	NA	
8	P4440	P4440H01	0	30	0.0	3.6	6.2	5.2	11.7	16.6	NA	
9	P5083	P5083H01	0	20	0.0	2.1	6.0	NA	NA	NA	NA	
10	P5083	P5083H02	20	40	0.0	1.7	6.9	NA	NA	NA	NA	
11	P5083	P5083H03	40	60	0.0	2.1	7.0	NA	NA	NA	NA	
12	P5085	P5085H01	0	70	12.6	10.5	NA	NA	NA	NA	NA	
13	P5085	P5085H02	70	100	1.8	5.1	NA	NA	NA	NA	NA	
14	P0364	P0364H02	20	50	0.0	3.0	7.1	6.1	7.7	25.0	NA	

Explore soil layers (horizons)

- Run the summary of dat_layers

```
summary(dat_layers)
```

Question 1: What's the **mean** value of organic matter content (**OM_perc**)?

Question 2: How many missing values (**NAs**) are there in bulk density (**Bulk_density_gcm3**)?

Question 3: Do you see anything unusual in the data? Any possible errors?

Import soil profile locations

```
# Import site-level data from a .csv table

dat_sites <- read.csv(file = "01-Data/site-level.csv")

# Explore the data

str(dat_sites)

summary(dat_sites)
```

```
> summary(dat_sites)
     ProfID                                soiltype         X                Y
 P0004  :   2   Fluvisol                        :476   Min.   :20.47   Min.   :40.69
 P0007  :   2   Cambisol                        :190   1st Qu.:21.34   1st Qu.:41.25
 P0008  :   2   Mollic Fluvisol                 :104   Median :21.62   Median :41.52
 P0011  :   2   Complex of Mollic and Umbric Leptosol: 99   Mean   :21.70   Mean   :41.56
 P0012  :   2   Chromic Luvisol on saprolite    : 80   3rd Qu.:22.19   3rd Qu.:41.89
 P0013  :   1   (Other)                         :982   Max.   :23.01   Max.   :42.36
 (Other):1952   NA's                            : 32
```

Remove duplicate points

- Test the data for duplicate points, based on columns 1 (ProfID), 3 (X) and 4 (Y).

```
# Remove duplicate profiles

test <- duplicated(dat_sites[c(1,3,4)])

summary(test)
```

Question: How many duplicate points are there??

- Remove duplicate points from the table

```
dat_sites <-  dat_sites[!duplicated(dat_sites[c(1,3,4)]),]
```

Merging data frames

- Now we can merge the horizon data with profile locations by the profile identifier (ProfID).

```
# Merge site-level data with soil layers (horizons)  data
dat <- merge (x=dat_sites, y=dat_layers, by="ProfID")
summary(dat)
names (dat)
```

```
> names (dat)
 [1] "ProfID"           "soiltype"           "X"                  "Y"                  "HorID"              "DepthFrom"
 [7] "DepthTo"          "CaCO3_perc"         "OM_perc"            "pH_H2O"             "pH_nKCl"            "P2O5_mg100g"
[13] "K2O_mg100g"       "Bulk_density_gcm3"  "Higroscopic_water"  "Stones_perc"        "Sand_perc"          "Silt_perc"
[19] "Clay_perc"
```

- The new data frame (dat) has both the information about profile (soil type, X, Y), and all the data from the horizons)

Selecting the data for modelling

- In this exercise we will model Soil Organic Carbon Stocks (OCS) for the topsoil layer of 30cm.
- $OCS = d * BD * SOC * CF$

  where: OCS = soil organic carbon stock [kg/m2], SOC = soil organic carbon [g /k g], d = depth class [m], BD = bulk density [kg/m3]

```
# Note that SOC is OM*1.724 (OM - organic matter)
dat <- data.frame(id=dat$ProfID,
                  X=dat$X,
                  Y=dat$Y,
                  soil=dat$soiltype,
                  top=dat$DepthFrom,
                  bottom=dat$DepthTo,
                  SOC=dat$OM_perc/1.724,
                  BLD=dat$Bulk_density_gcm3,
                  CRF=dat$Stones_perc)
```

Explore and clean the SOC data

```
# Explore and clean the SOC data

summary(dat$SOC)
```

```
> summary(dat$SOC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.6961  1.0441  1.6574  1.7981 48.6079     152
```

- We cannot use points where SOC is **NA** or **0**. They should be removed:

```
dat <- dat[!is.na(dat["SOC"]),] # remove NA values

dat <- dat[dat$SOC>0,] # remove 0 values
```

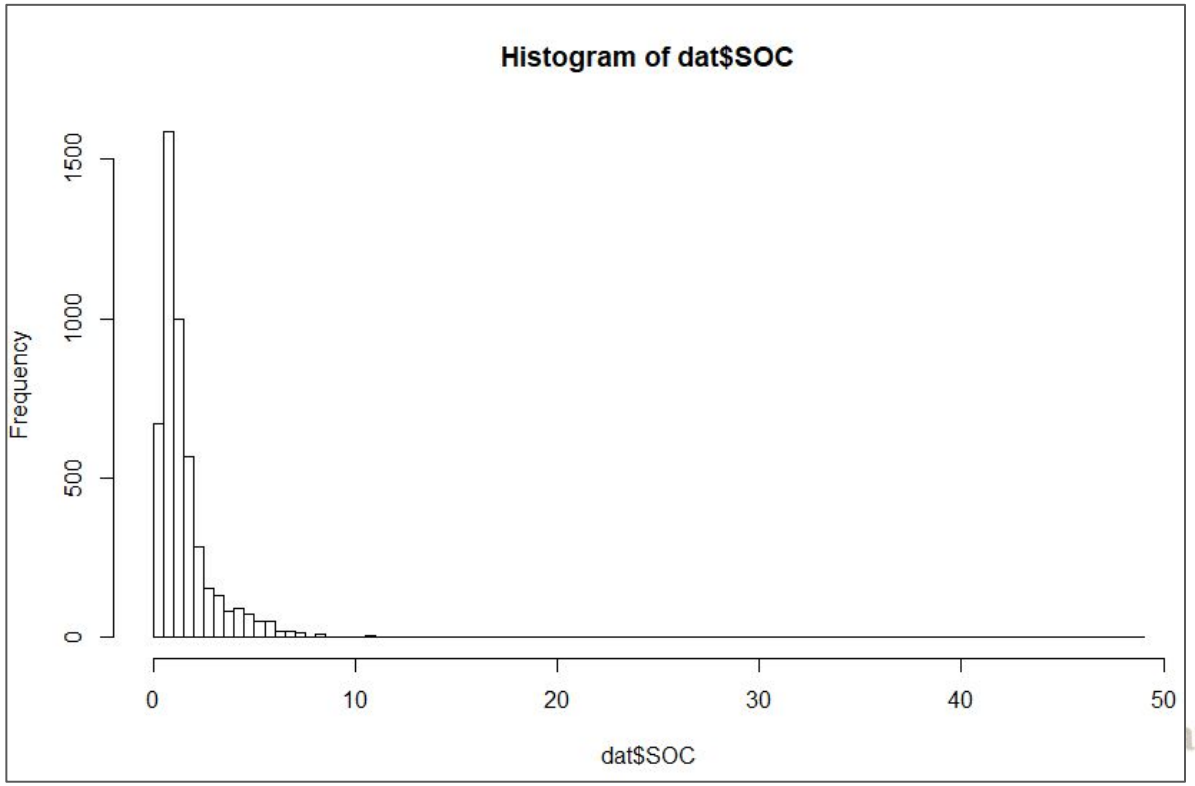
Outliers

- **Outliers** are data values that differ greatly from the majority of a dataset;
- They can be **errors**, or they can be correct, but **unusual values**
- Outliers can be formally defined using **interquartile range (IQR)** (difference between 1st and 3rd quartiles). If the value is 1.5 x IQR lower than 1st quartile, or 1.5 x IQR higher than 3rd quartile, then such value is an **outlier**.
- **Histograms** can help to visually identify outliers
- **Box-plot** is the best tool to identify outliers (visually and automatically)

Identifying outliers

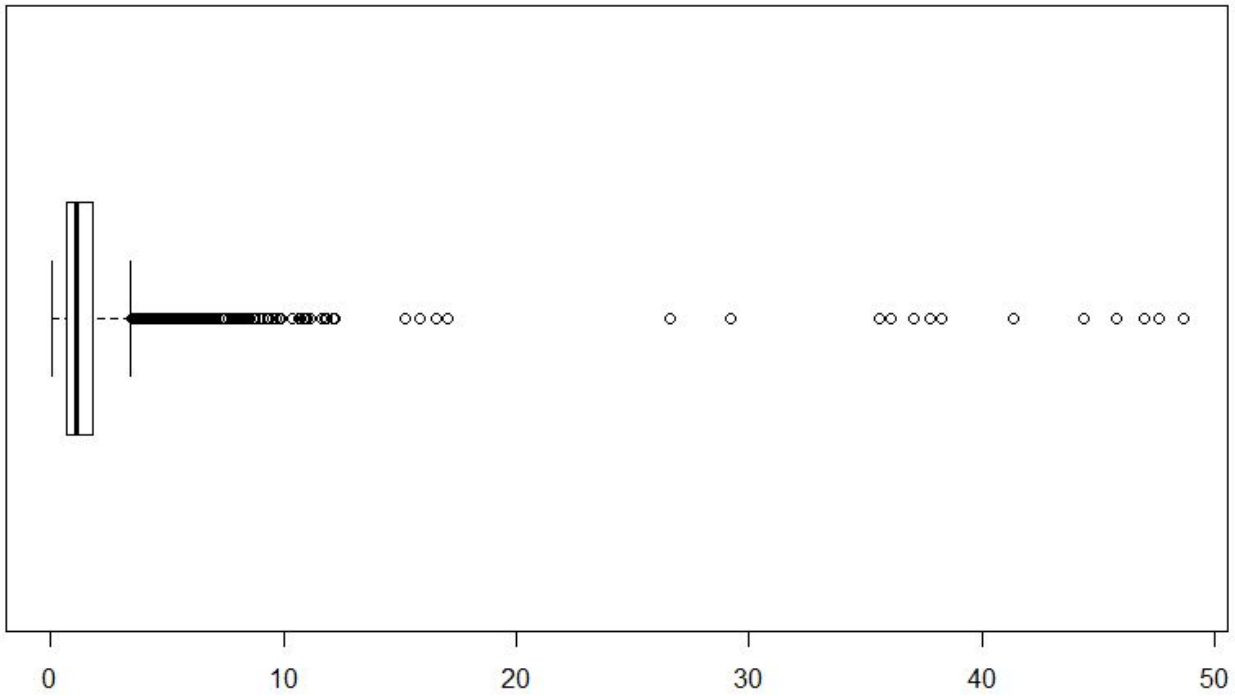
```
# Explore SOC data, identify outliers

summary(dat$SOC)


hist(dat$SOC, breaks=100)
```

```
> summary(dat$SOC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0580  0.6961  1.1021  1.6602  1.7981 48.6079
```



Identifying outliers with boxplot

```
boxplot(dat$SOC, horizontal=TRUE)
```



Question:
Can SOC be higher than 15% (OM>25%)?

Exploring outliers

- Let’s explore, which soil types have very high SOC values

```
summary(dat[which(dat$SOC>15),])
```

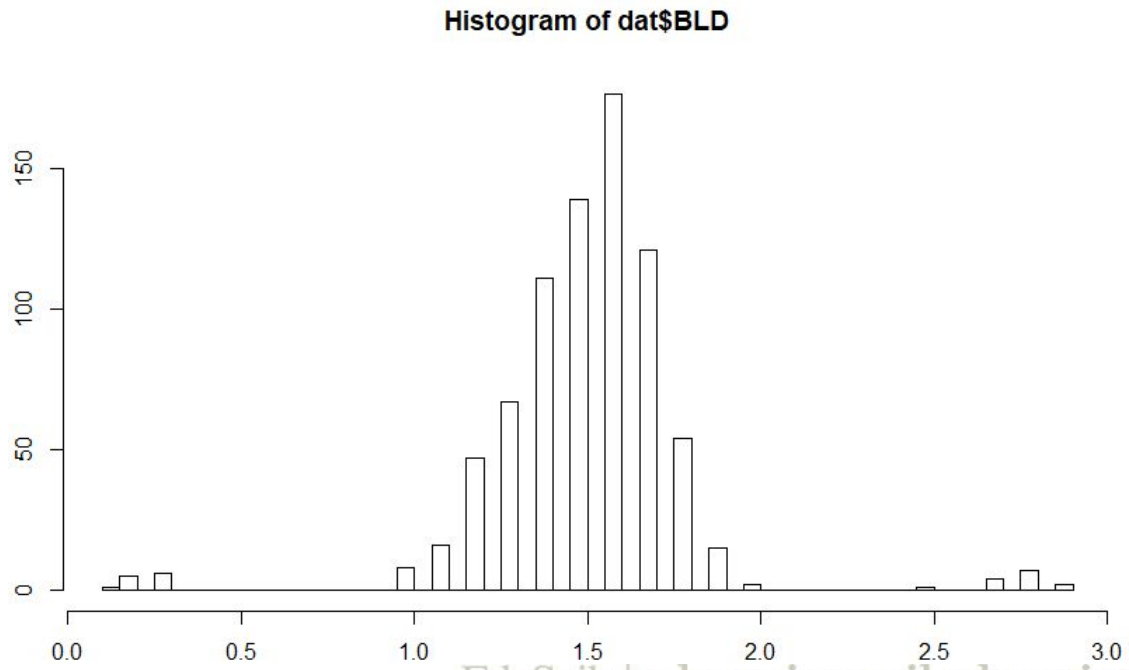
```
> summary(dat[which(dat$SOC>15),])
       id          X                Y                                                                soil
 P5910  :3   Min.   :20.58   Min.   :41.12   Histosol                                                    :12
 0.000
 P5912  :3   1st Qu.:21.00   1st Qu.:41.68   Cambisol                                                    : 2
 0.000
 P5908  :2   Median :21.12   Median :42.05   Chromic Luvisol on hard limestones                          : 1
 4.000
 P5909  :2   Mean   :21.21   Mean   :41.87   Complex of Mollic and Umbric Leptosol                       : 1
 9.526
 P5911  :2   3rd Qu.:21.12   3rd Qu.:42.05   Complex of Rendzic Leptosol and Chromic Luvisol on hard limestones: 1
16.000
 P0004  :1   Max.   :22.82   Max.   :42.05   Fluvisol                                                    : 1
40.000
 (Other):6                                   (Other)                                                     : 1
```

```
# We will remove all non-histosols with atypically high SOC as outliers
dat <- dat[!( dat$SOC>15 & dat$soil!='Histosol'),]
```

Explore bulk density data

```
# Explore bulk density data, identify outliers

summary(dat$BLD)

dat$BLD[dat$BLD==0]<-NA    # bulk density cannot be 0, it must be NA

hist(dat$BLD, breaks=50)
```

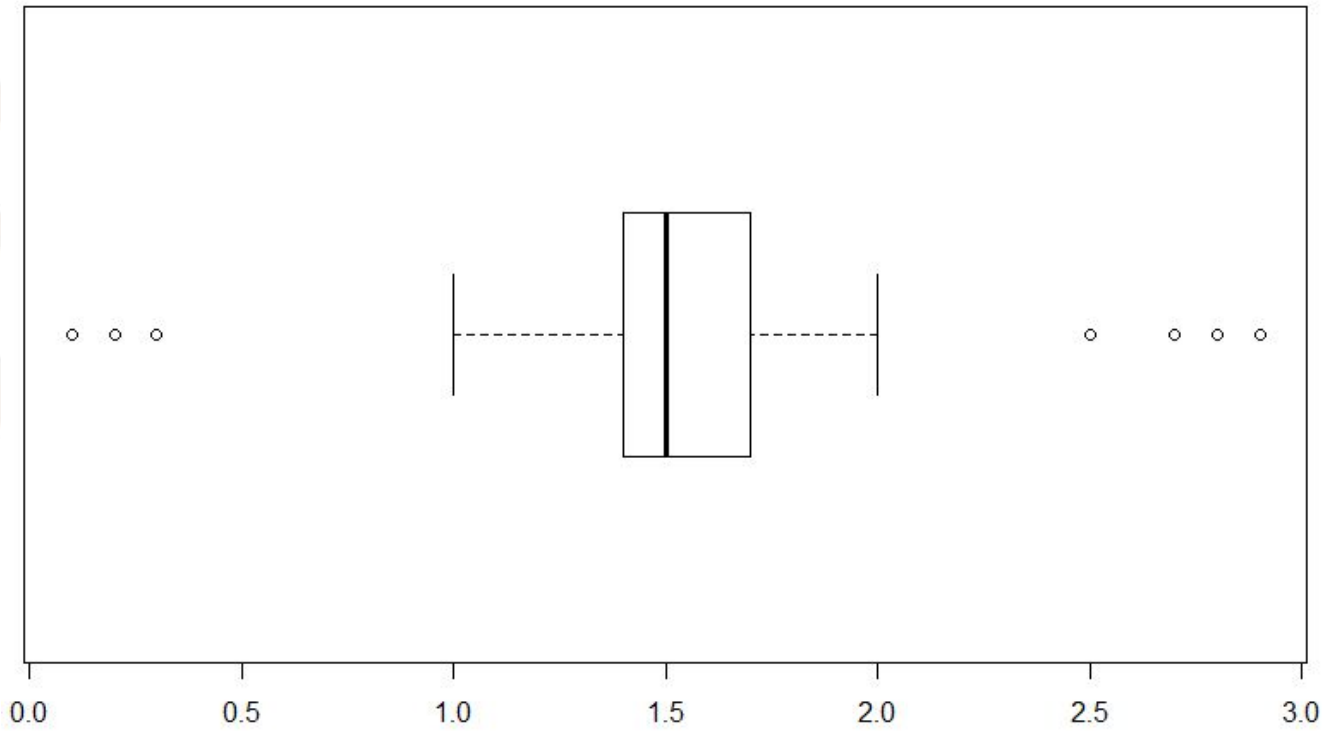
```
> summary(dat$BLD)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  0.000   1.400   1.500   1.518   1.700   2.900    4070
```



Explore bulk density data

```
boxplot(dat$BLD, horizontal=TRUE)$out
```

```
> boxplot(dat$BLD, horizontal=TRUE)$out
 [1] 2.9 2.8 2.7 2.8 2.8 2.9 2.8 2.8 2.8 2.5 2.7 2.8 2.7 2.7 0.2 0.2 0.3 0.2 0.3 0.2 0.3 0.3 0.1 0.3 0.3 0.2
```



Clean bulk density data

- BLD values higher than 2.5 g·cm are not typical for fine earth, and most likely correspond to coarse fragments;
- These values should be **removed**

```
dat <- dat[!(dat$BLD>=2.5 & !is.na(dat$BLD)),]
```

- BLD values lower than 1 g·cm3 can only correspond to organic soils (Histosols)
- We should check if all low BLD values correspond to Histosols

```
summary(dat[dat$BLD<1,])
```

```
dat <- dat[ !(dat$BLD<1 & dat$soil!='Histosol'),]
```

```
       id              X               Y                soil          top             bottom
 SOC
 P5910  :   3   Min.   :21.12   Min.   :42.05   Histosol      :  12   Min.   : 0.00   Min.   :12.00
.   :29.18
 P5912  :   3   1st Qu.:21.12   1st Qu.:42.05   Albic Luvisol :   0   1st Qu.: 0.00   1st Qu.:20.00
Qu.:36.82
 P5908  :   2   Median :21.12   Median :42.05   Aric Anthrosol :   0   Median :12.00   Median :32.00
```

Clean coarse fragments data

```
# Explore and clean coarse fragments data

summary(dat$CRF)

hist(dat$CRF, breaks=50)

# Remove outliers automatically, using boxplot

out <- boxplot(dat$CRF, horizontal=TRUE)$out

dat <- dat[!(dat$CRF %in% out),]

# The points without CRF, should have 0 values

dat$CRF[is.na(dat$CRF)]<-0
```

Pedotransfer functions

- We have many points with missing **bulk density**, in this case it can be estimated using **pedotransfer functions**
- Pedotransfer functions (rules) are **empirical relationships** between the certain (difficult to obtain) soil properties and other (more easily obtainable) soil properties available, for example, from soil surveys.
- **Pedotransfer rules** are based on empirical data from case-studies. If applied for soils which are different from the case study, the function should be **checked on local data** first.

Pedotransfer functions

- Let's write an R function to compare different pedotransfer rules, and choose the one that fits our local data
- First, we select a **testing subset** of our data, where we **do have** BLD

```
# Select the best fitting pedotransfer function

BD_test <- dat[is.na(dat$BLD)==FALSE,]

estimateBD <- function(SOC, method="Saini1996"){
  OM <- SOC * 1.724
  if(method=="Saini1996"){BD <- 1.62 - 0.06 * OM}
  if(method=="Drew1973"){BD <- 1 / (0.6268 + 0.0361 * OM)}
  if(method=="Jeffrey1979"){BD <- 1.482 - 0.6786 * (log(OM))}
  if(method=="Grigal1989"){BD <- 0.669 + 0.941 * exp(1)^(-0.06 * OM)}
  if(method=="Adams1973"){BD <- 100 / (OM / 0.244 + (100 - OM)/2.65)}
  if(method=="Honeyset_Ratkowsky1989"){BD <- 1/(0.564 + 0.0556 * OM)}
  return(BD)
}
```

Selecting a pedotransfer function

- We run different pedotransfer functions on the testing data

```
# Estimate BLD for a subset using the pedotransfer functions

BD_test$Saini <- estimateBD(BD_test$SOC, method="Saini1996")

BD_test$Drew <- estimateBD(BD_test$SOC, method="Drew1973")

BD_test$Jeffrey <- estimateBD(BD_test$SOC, method="Jeffrey1979")

BD_test$Grigal <- estimateBD(BD_test$SOC, method="Grigal1989")

BD_test$Adams <- estimateBD(BD_test$SOC, method="Adams1973")

BD_test$Honeyset_Ratkowsky <- estimateBD(BD_test$SOC, method="Honeyset_Ratkowsky1989")
```

Selecting a pedotransfer function

- Now we can explore summaries of the results

```
# Compare results

# Observed values:

summary(BD_test$BLD)

# Predicted values:

summary(BD_test$Saini)

summary(BD_test$Drew)

summary(BD_test$Jeffrey)

summary(BD_test$Grigal)

summary(BD_test$Adams)

summary(BD_test$Honeyset_Ratkowsky)
```

Selecting a pedotransfer function

```
> # Observed values:
> summary(BD_test$BLD)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   1.400   1.500   1.501   1.600   2.000
>
> # Predicted values:
> summary(BD_test$Saini)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -3.408   1.464   1.518   1.434   1.554   1.614
> summary(BD_test$Drew)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2738  1.3876  1.4531  1.4169  1.5004  1.5863
> summary(BD_test$Jeffrey)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.5231  0.8336  1.1219  1.0950  1.4173  3.0445
> summary(BD_test$Grigal)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6752  1.4741  1.5188  1.4916  1.5499  1.6044
> summary(BD_test$Adams)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2861  2.1092  2.2696  2.2015  2.3907  2.6241
> summary(BD_test$Honeyset_Ratkowsky)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1915  1.4113  1.5186  1.4730  1.5996  1.7557
```

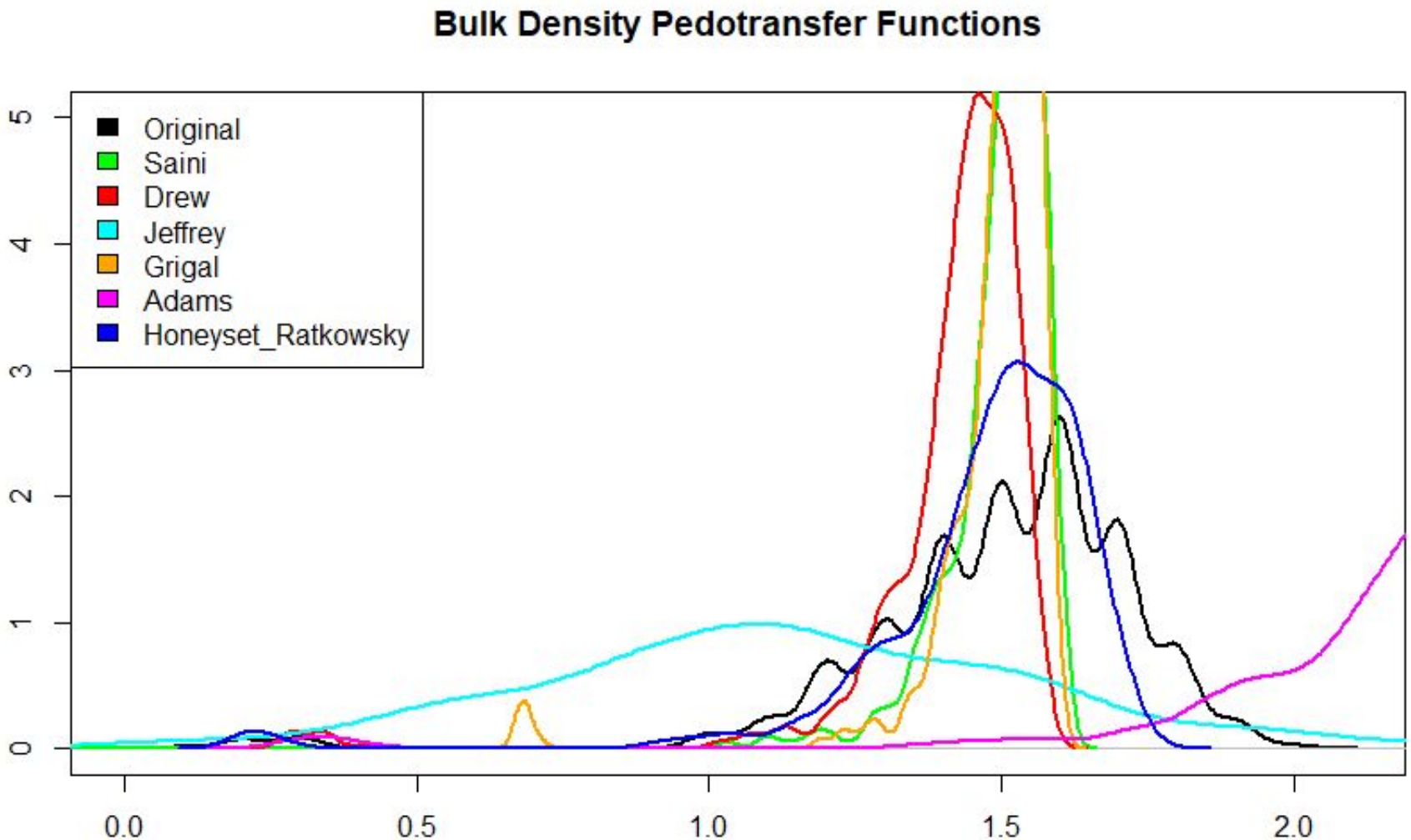
Question 1:
Which functions have impossible or unlikely values?

Question 2:
Which functions are closer to the observed data?

Selecting a pedotransfer function

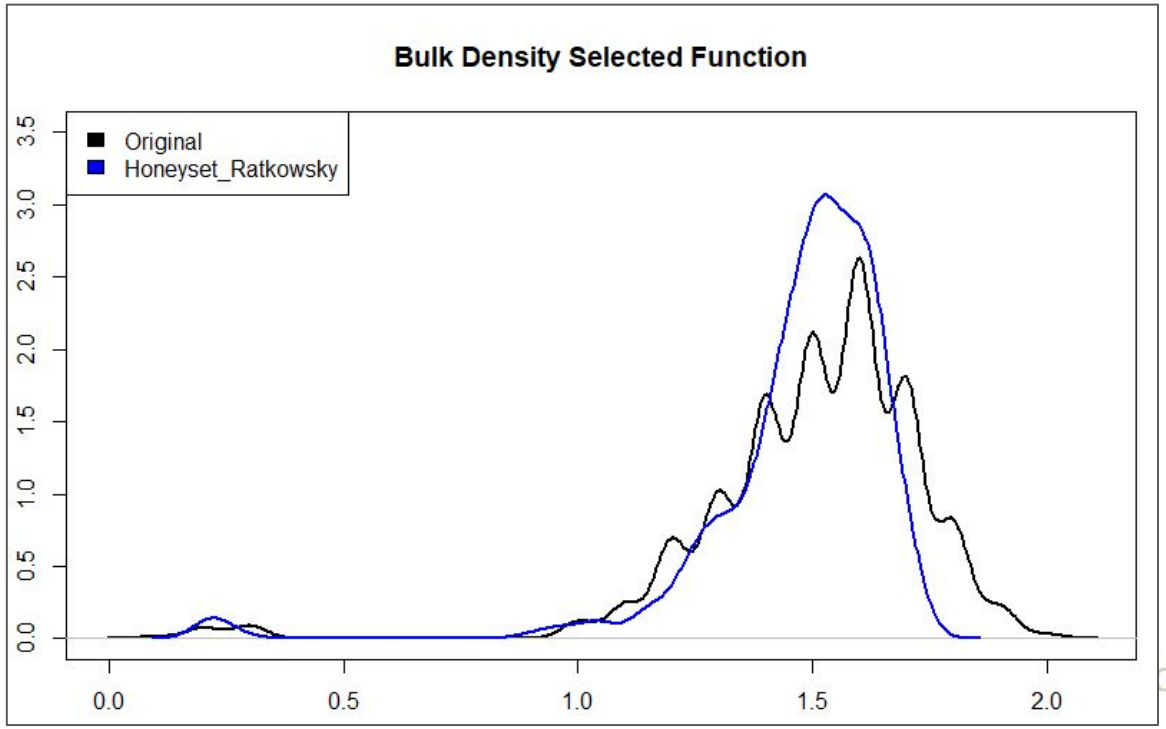
- Let's compare graphically!

```
# Compare data distributions for observed and predicted BLD
plot(density(BD_test$BLD),type="l",col="black", ylim=c(0,5), lwd=2, main="Bulk Density
Pedotransfer Functions")
lines(density(BD_test$Saini),col="green", lwd=2)
lines(density(BD_test$Drew),col="red", lwd=2)
lines(density(BD_test$Jeffrey),col="cyan", lwd=2)
lines(density(BD_test$Grigal),col="orange", lwd=2)
lines(density(BD_test$Adams),col="magenta", lwd=2)
lines(density(BD_test$Honeyset_Ratkowsky),col="blue", lwd=2)
legend("topleft",legend = c("Original", "Saini", "Drew", "Jeffrey", "Grigal",
"Adams","Honeyset_Ratkowsky"), fill=c("black", "green", "red", "cyan", "orange","magenta",
"blue"))
```



Selecting a pedotransfer function

```
# Plot the Selected function again
plot(density(BD_test$BLD),type="l",col="black", ylim=c(0,3.5), lwd=2, main="Bulk Density
Selected Function")
lines(density(BD_test$Honeyset_Ratkowsky),col="blue", lwd=2)
legend("topleft",legend = c("Original", "Honeyset_Ratkowsky"), fill=c("black", "blue"))
```



Estimating bulk density

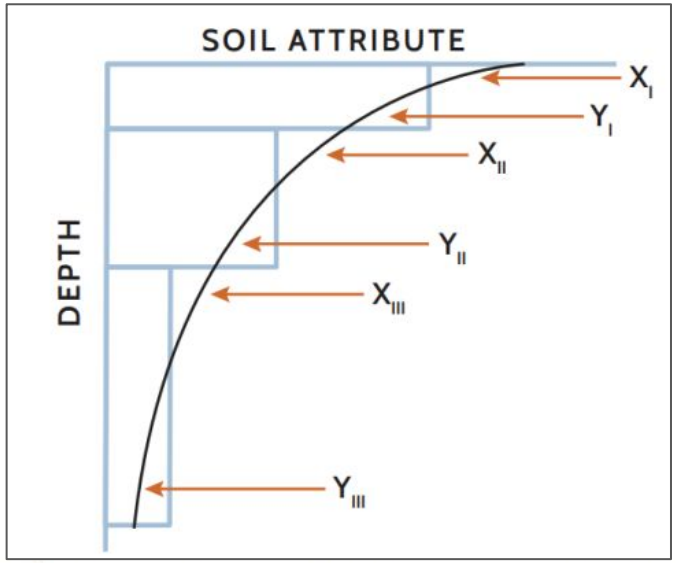
```
# Estimate BLD for the missing points with the selected function
dat$BLD[is.na(dat$BLD)] <- estimateBD(dat[is.na(dat$BLD),]$SOC,
method="Honeyset_Ratkowsky1989")

# Explore the results
summary(dat$BLD)
plot(density(BD_test$BLD),type="l",col="black", ylim=c(0,3.5), lwd=2, main="Bulk Density
Gap-Filling")
lines(density(dat$BLD),col="green", lwd=2)
legend("topleft",legend = c("Original", "Original+Estimated"), fill=c("black", "green"))
```

Question:
Does the gap-filled bulk density have a similar distribution to the original one?

Estimating target depth

- Profile data has soil parameters measured for **every horizon** (depth layer)
- We need to estimate mean value for target depth: **0-30cm**
- For that we can use **equal-area splines**. This technique is based on fitting continuous depth functions for modeling the variability of soil properties with depth.
- We can use equal-area splines in with the package **aqp** - "algorithms for quantitative pedology", which was specifically designed to work with soil data.

top	bottom	SOC	BLD	CRF
23	34	1.6241299	1.3895064	35
0	23	2.4941995	1.2452060	31
8	29	2.0881671	1.3086265	29
29	48	1.8561485	1.3478542	27
0	8	2.8422274	1.1955430	23
10	30	1.6821346	1.3788539	24
30	65	1.6821346	1.3788539	27



Estimating target depth

```
# Promote the data frame to ‘soil profile collection’ data type
library(aqp)
depths(dat) <- id ~ top + bottom
site(dat) <- ~ X + Y
coordinates(dat) <- ~ X + Y

# Estimate 0-30 standard horizon using mass preserving splines
library(ithir)

try(SOC <- ea_spline(dat, 'SOC', d = t(c(0,30))))
try(BLD <- ea_spline(dat, 'BLD', d = t(c(0,30))))
try(CRFVOL <- ea_spline(dat, 'CRF', d = t(c(0,30))))

dat <- data.frame(id = dat@site$id,
                  Y = dat@sp@coords[,2],
                  X = dat@sp@coords[,1],
                  SOC = SOC[["harmonised"]][["0-30 cm"]],
                  BLD = BLD[["harmonised"]][["0-30 cm"]],
                  CRFVOL = CRFVOL[["harmonised"]][["0-30 cm"]])
summary(dat)
```

Calculating organic carbon stock

- *OCS = d * BD * SOC * CF*

  where: OCS = soil organic carbon stock [kg/m2], SOC = soil organic carbon [g / kg], d = depth class [m], BD = bulk density [kg/m3]

```
# Estimate Organic Carbon Stock
# SOC must be in g/kg (% * 10)
# BLD in kg/m3 (*1000)
# CRF in percentage
library(GSIF)
OCSKGM <- OCSKGM(ORCDRC = dat$SOC*10, BLD = dat$BLD*1000, CRFVOL = dat$CRF, HSIZE = 30)
# Convert Organic Carbon Stock from kg/m3 to t/ha
dat$OCS <- OCSKGM*10
# Explore calculated SOC stocks
summary(dat$OCS)
```

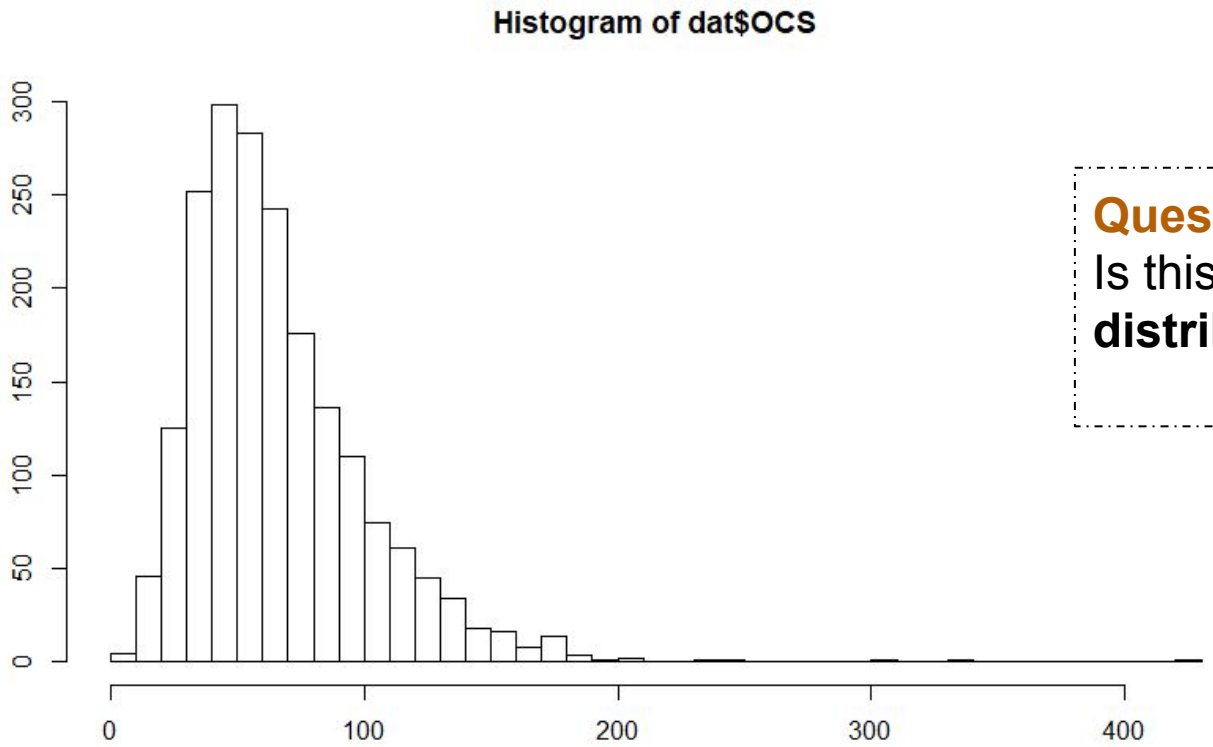
```
> summary(dat$OCS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  7.643  42.214  59.034  66.199  82.390 425.379
```

```
OCSKGM <-function (ORCDRC, BLD = 1400, CRFVOL = 0, HSIZE, ORCDRC.sd = 10,
        BLD.sd = 100, CRFVOL.sd = 5, se.prop = TRUE)
{
  if (any(ORCDRC[!is.na(ORCDRC)] < 0) | any(BLD[!is.na(BLD)] <
                                    0) | any(CRFVOL[!is.na(CRFVOL)] < 0)) {
    warning("Negative values for 'ORCDRC', 'BLD', 'CRFVOL' found")
  }
  OCSKG <- ORCDRC/1000 * HSIZE/100 * BLD * (100 - CRFVOL)/100
  if (se.prop == TRUE) {
    if (any(ORCDRC.sd[!is.na(ORCDRC.sd)] < 0)) {
      ORCDRC.sd = ifelse(is.na(ORCDRC.sd) | ORCDRC.sd <
                         0, 0, ORCDRC.sd)
      warning("Replacing negative values for 'ORCDRC.sd'")
    }
    if (any(BLD.sd[!is.na(BLD.sd)] < 0)) {
      BLD.sd = ifelse(is.na(BLD.sd) | BLD.sd < 0, 0, BLD.sd)
      warning("Replacing negative values for 'BLD.sd'")
    }
    if (any(CRFVOL.sd[!is.na(CRFVOL.sd)] < 0)) {
      CRFVOL.sd = ifelse(is.na(CRFVOL.sd) | CRFVOL.sd <
                         0, 0, CRFVOL.sd)
      warning("Replacing negative values for 'CRFVOL.sd'")
    }
    OCSKG.sd <- 1e-07 * HSIZE * sqrt(BLD^2 * (100 - CRFVOL)^2 *
                                     ORCDRC.sd^2 + ORCDRC^2 * (100 - CRFVOL)^2 * BLD.sd^2 +
                                     ORCDRC^2 * BLD^2 * CRFVOL.sd^2)
    attr(OCSKG, "measurementError") <- signif(OCSKG.sd,
                                              3)
    attr(OCSKG, "units") <- "kilograms per square-meter"
  }
  return(OCSKG)
}
```

Checking the data distribution

- Let’s check the distribution of our final layer

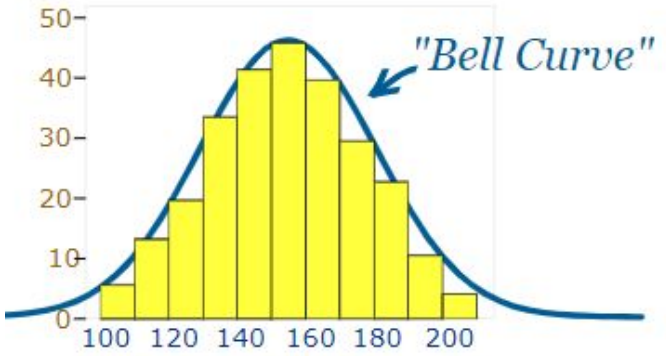
```
hist(dat$OCS, breaks = 50)
```



Question:
Is this a **normal distribution**?

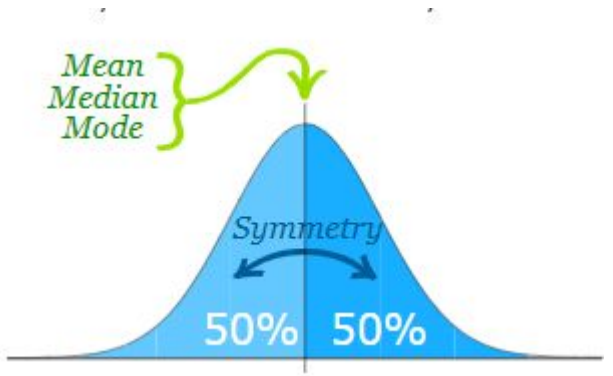
Normal Distribution

- **Normal distribution**, also known as the **Gaussian distribution**, is a probability distribution that is **symmetric** about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.



The **Normal Distribution** has:

- mean = median = mode
- symmetry about the center
- 50% of values less than the mean and 50% greater than the mean



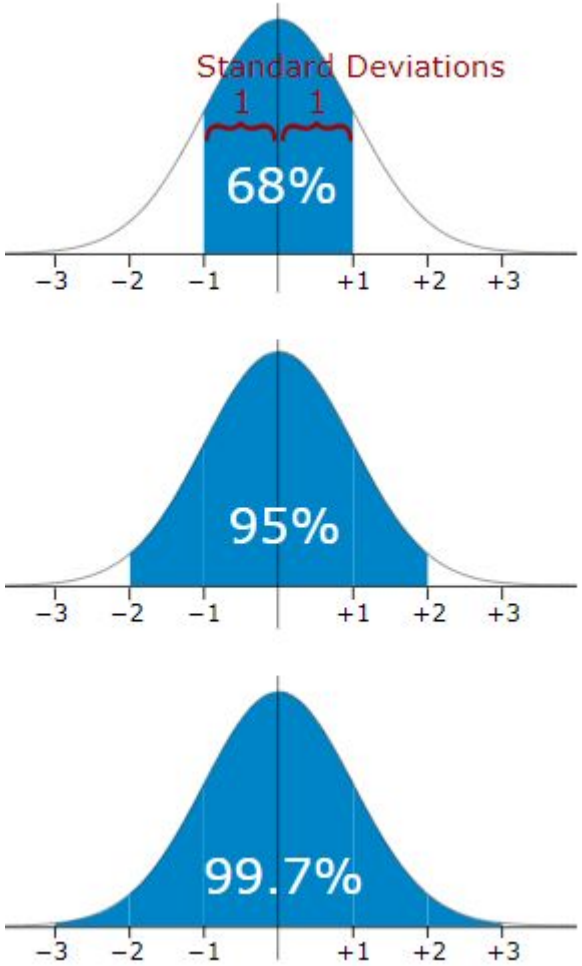
Normal Distribution

- **Standard Deviation** (SD) is a measure that is used to quantify the amount of variation or dispersion of the data
- SD is the square root of the **Variance**. Variance is the average of the squared differences from the Mean.

Normal distribution has:

- 68% of values are within 1 SD of the mean
- 95% of values are within 2 SD of the mean
- 99.7% of values are within 3 SD of the mean

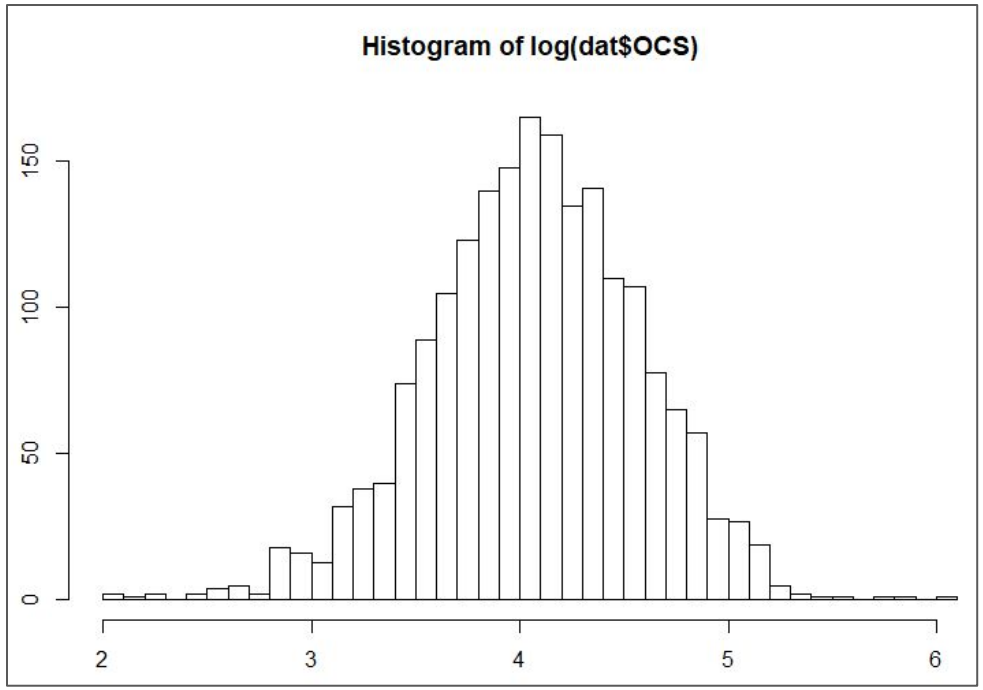
In **statistical analyses** it is often **assumed** that the data has normal distribution. If it does not, it may be useful to **transform** the data.



EduSoils | e-learning soil educational platform

log-transforming the data

```
# Check if log-transformation improves # the data
distribution

hist(log(dat$OCS), breaks = 50)


# Add a new column for
# log-transformed carbon stocks
dat$OCSlog <- log(dat$OCS)


# Save the final table in a .csv file
write.csv(dat, "02-Outputs/dataproc.csv", row.names
= FALSE)
```



Splitting the data

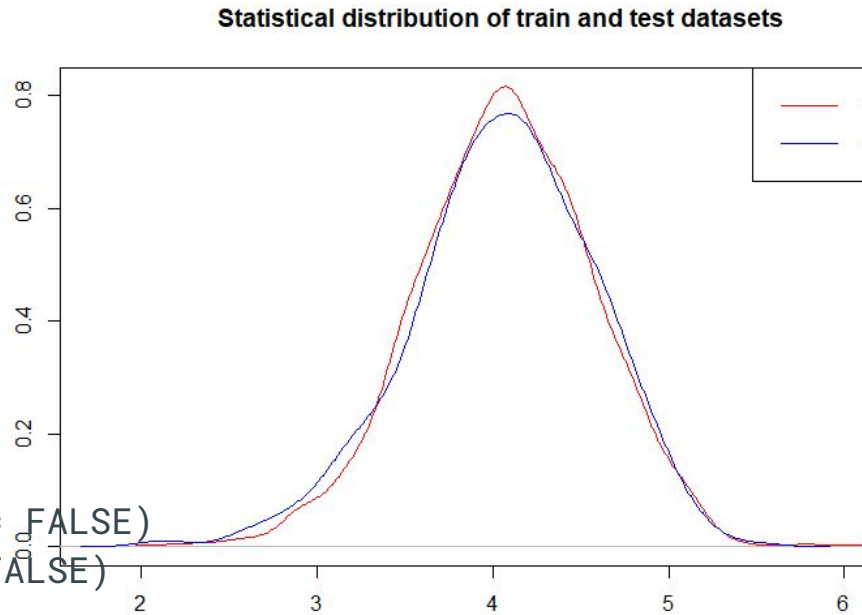
- When predicting soil properties, it is useful to have part of the data **not used** in the prediction model, so that it can be used to **validate** (check the accuracy) the modelling result
- To have such data for **validation**, we will randomly split the dataset in 2 parts:
    - data for **training** the model
    - data for **testing** the model

```
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,]
```

Comparing 'train' and 'test' data

```
# Check if both datasets
# have similar distributions
summary(train$OCS)
summary(test$OCS)

plot(density (train$OCSlog), col='red', main='Statistical
distribution of train and test datasets')
lines(density(test$OCSlog),col='blue')
legend('topright', legend=c("train", "test"), col=c("red",
"blue"), lty=1, cex=1.5)
```

```
> 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
```




```
# Save the 'train' and 'test' datasets
write.csv(train,file="02-Outputs/dat_train.csv", row.names = FALSE)
write.csv(test,file="02-Outputs/dat_test.csv", row.names = FALSE)
```

Conclusions

Don't forget to save your script!

Questions:

- What are main steps of preparing soil profile data for digital soil mapping?
- What should you check for?

Conclusions

Main steps for preparing data:

- Merging horizons with profile locations;
- Cleaning the data (check for errors);
- Filling the gaps in the data (pedotransfer functions);
- Calculating the target depth (spline function);
- Transforming the data (if distribution is not normal);
- Splitting the data (for validating the model)

Conclusions

Things to check for:

- Duplicate points
- NA values
- 0 values
- Outliers
- Validity of pedotransfer functions
- Data distribution