



# ■ RGeostats

D. Renard

N. Desassis

## o Principles of the course

- RGeostats is used to practice the geostatistical concepts:
  - Mapping Geoslib commercial library
  - Writing complex scripts
- RGeostats is based on classes and methods (commonly used in the R language)
- Main authors:
  - H. Beucher (\*)
  - N. Bez (\*\*)
  - N. Desassis (\*)
  - D. Renard (\*)
  - F. Ors (\*)

(\*) Center of Geosciences - Mines ParisTech

(\*\*) Institut de recherche pour le développement

## o Basics

➤ Some code:

```
4+5
```

```
log(1)
```

```
log(2)
```

```
a
```

Why an error

```
a=1
```

```
b=5
```

```
d=a+b
```

```
d
```

```
d=log(a)
```

```
d
```

In **red**, standard R commands

## o Demonstration

- Demonstrate the Data Base using information from the ASCII file (provided to the user as auxiliary file):

`Scotland_Temperatures.csv`

- The file is loaded as follows

```
temperatures = read.csv("Scotland_Temperatures.csv",header=T,na="MISS")
```

The file contains missing information.

- Temperatures is now an item (object) of your workspace:

```
ls()
```

- It corresponds to a data.frame :

```
class(temperatures)
```

- To check its contents:

```
temperatures
```

In `red`, standard R commands

## o Introduction

- **db**: numerical data base
- **vario** (**vardir**): experimental variograms
- **model** (**melem**): variogram model
- **neigh**: neighborhood
- **anam**: gaussian anamorphosis
- **polygon** (**polyset**): 2-D polygonal shapes
- **limits**: bounds
- **rule**: lithotype rule for thresholds (PGS)
- **tokens**: Object definition (Boolean)

## o Discover the Db class

- Turn a data.frame into a Db

```
dat=db.create(temperatures)
```

- Check its contents:

```
dat or print(dat) or db.print(dat)
```

```
print(dat,flag.stats=T)
```

```
print(dat,flag.stats=T,name=4)
```

- Specify the variable roles (locators):

```
dat=db.locate(dat,2:3,"x")
```

```
dat=db.locate(dat,4:5,"z")
```

- Get more information about commands:

```
?db.locate
```

```
args(db.locate)
```

```
class?db
```

In blue, RGeostats commands

## o Perform selection

➤ Creating a selection based (230):

- Selecting samples such that Latitude < 1000
- Using field names

```
dat=db.sel(dat,Latitude<1000)
```

```
dat
```

➤ Update the previous selection so as to keep samples (141):

- With Latitude < 1000
- With elevation < 50m

```
dat = db.sel(dat,Elevation > 50,combine="and",flag.replace=T )
```

➤ Cancel the selection (although keeping the variable in the Db) (236)

```
dat=db.sel(dat)
```

➤ Activate an already existing selection (87)

```
dat=db.sel(dat,nameold="sel" )
```

## o Graphic representation

- Plot the data

```
dat=db.sel(dat)
```

```
plot(dat)
```

- Plotting options:

```
?db.plot
```

- Proportional representation on isometric view (blue symbols):

```
plot(dat,scale=1,pch=21,col="blue")
```

- Representation on isometric view with symbols colored as a function of the Elevation

```
dat
```

```
plot(dat,name.post=4,scale=1,pch=21,col=rainbow(10))
```



## o Grid organization

- Create a dummy 2-D grid covering the bottom part of the field with a square mesh of 50m

```
gdum = db.create(dx=c(50,50),nx=c(7,11),x0=c(100,500))
```

- Overlay the data and the grid nodes

```
plot(dat,scale=1)
```

```
plot(gdum,name.post=1,add=T,pch=21,col="blue")
```

- Load an already existing grid

```
data(Exdemo_Scotland_Elevations)
```

```
grid = Exdemo_Scotland_Elevations
```

- Representation of the grid (raster) and the Temperature overlaid

```
plot(grid,scale=1)
```

```
plot(dat,name.post=5,pch=21,col="black",add=T,cex=.5)
```

## o Define a Polygon interactively

- Draw the grid of elevations of Scotland

```
plot(grid,scale=1)
```

- Digitize a polygon with several components (polysets):

```
pol = polygon.digit()
```

- Check the contents of the polygon

```
pol
```

```
polygon.print(pol,flag.print=1)
```

- Use the polygon to mask part of the grid

```
grid=db.polygon(grid,pol)
```

```
plot(grid,scale=1)
```

## o Assessors

- Assessors is a specific syntax used to reach some elements of the Db Class in an abbreviated manner:

- Get the information about the **slots**:

`dat$nech`

`dat$nactive`

`dat$ndim`

- Get the information row-columns:

`dat[,5]`

`dat[20:30,4:5]`

- Examples of usage:

`hist(dat[,5],nclass=20)`

`plot(dat[,4], dat[,5])`