Exercise 3L1. Landslide susceptibility assessment using statistical method

Expected time:	3 hours
Data:	data from subdirectory:Riskcity_exercise/exercise03L1/data
Objectives:	This exercise shows you how carry out a basic bivariate statistical landslide susceptibility assessment, using a limited number of factor maps, and only one landslide type. The method used is the information value method, one of the simplest methods which can be easily implemented in GIS. The use of scripts is also demonstrated. The output map is validated using the success rate method.

Some background information:

In this exercise we will generate a landslide susceptibility map, using a basic, but useful, statistical method, called hazard index method. This method is based upon the following formula:

$$W_i = \ln\left(\frac{\text{Densclas}}{\text{Densmap}}\right) = \ln\left(\frac{\frac{\text{Area(Si)}}{\text{Area(Ni)}}}{\frac{\sum \text{Area(Si)}}{\sum \text{Area(Ni)}}}\right)$$

where,

Wi = the weight given to a certain parameter class (e.g. a rock type, or a slope class).

Densclas = the landslide density within the parameter class.

Densmap = the landslide density within the entire map.

Area(Si) = area, which contain landslides, in a certain parameter class.

Area(Ni) = total area in a certain parameter class.

The method is based on map crossing of a landslide map with a certain parameter map. The map crossing results in a cross table, which can be used to calculate the density of landslides per parameter class. A standardization of these density values can be obtained by relating them to the overall density in the entire area. The relation can be done by division or by subtraction. In this exercise the landslide density per class is divided by the landslide density in the entire map. The natural logarithm is used to give negative weights when the landslide density is lower than normal, and positive when it is higher than normal. By combining two or more maps of weight-values a hazard map can be created. The hazard map value is obtained by simply adding the separate weight-values. We are only using two factor maps in this exercise: Lithology and Slope, as the aim is to learn the procedure. In reality many possible factor maps should be evaluated.

Statistical landslide assessment:

There are two main types of statistical landslide susceptibility assessment: multi-variate and bivariate methods. Both require a landslide map which should contain only 1 type of landslides. Each landslide type or failure mechanism has its own combination of causal factors. The aim is to be able to separate the various types as best as possible. Also you should only use the scarp areas, and not the accumulation areas, because there the factors are considerably different.

Input data

In this exercise we will use the landslide inventory map Landslide_ID, which we have used in the previous exercises and a number of factor maps, listed in the table below.

Name	Type	Meaning
Factor data		
Slope_cl	Raster	Slope class map
Aspect_cl	Raster	Slope direction map (with classes)
Lithology	Raster	Lithological map
Soildepth	Raster	Soildepth map
Landuse	Raster	Landuse map
River_dis	Raster	Distance from rivers
Road_dis	Raster	Distance from roads
Landslide data		
Landslide_ID	Raster map	Points within each of the interpreted landslides with associated attribute table
Landslide_ID	Table	Attribute table with information on the landslides in the area.
Other data		
Building_map_segments	Segment map	Boundary lines of the buildings in the area.
High_res_image	Raster	High resolution image of the study area.

Landslide susceptibility versus hazard:
A landslide susceptibility map indicates the relative susceptibility of the terrain for the occurrence of landslides. It only has a spatial component. A landslide hazard map also contains information on the temporal probability of occurrence. Most so-called landslide hazard maps are in fact only landslide susceptibility maps, as it is very difficult to obtain sufficient temporal landslide information for a temporal probability assessment.

DFDX filter:
It calculates the first derivative in x-direction (df/dx) per pixel. The values in the matrix are:
1 -8 0 8 -1
Gain factor = 1/12 = 0.0833333
DFDY filter:
It calculates the first derivative in y-direction (df/dy) per pixel.

In this exercise the method for landslide susceptibility assessment is made by using only one factor map: **Slope_cl** (slope class map). The landslides are stored in the map **Landslide_ID**, which contains information on several attributes.

☞

- Open the map **High_res_image** and overlay the **landslide_ID** map. Also open several of the factor maps and check their contents.

Along side the landslide map you also have two parameter maps: Lithology (geological units) and Slope (slope angles).

For experienced ILWIS users:

Creating slope and aspect maps.

☞ For those interested in the procedure for the generation of slope class and slope aspect maps, you can follow the following procedure :

- Create a DTM by interpolation of the contours (*Operations / Interpolation / Contour interpolation*).
- To calculate height differences in X-direction: start the Filter operation, select the Digital Elevation Model as the input map and select linear filter dfdx. Call the output map for example DX. Do the same for the y direction, using filter dfdy. Name output: DY.
- Calculate the slope in degrees using the formula in MapCalc:

SLOPEDEG = RADDEG(ATAN(HYP(DX,DY)/ PIXSIZE(DEM)))

- Calculate the aspect in degree using the following formula:

ASPECTD = RADDEG(ATAN2(DX,DY) + PI)

- The map **Slope** still needs to be classified into classes (*File/Create/ Domain*). Make a class (don't forget to indicate the option *group*) domain **Slopecl** , and add the slope classes you want to differentiate. E.g. you can make classes of 10 degree each.
- Select from the main window: *Operations / Image Processing /Slicing*. Select the raster map Slope, and the domain **Slopecl**. Name the output map **Slopecl**. Same procedure for Aspect.

Conditional statements:
IFF(a, b, c)
If a is true, then return b, else return c.
IFF returns:
if a=true, b is returned;
if a=false, c is returned;
if a=undefined, undefined is returned.
The amount of nested IFF statements is unlimited
When the definition symbol = is used, a dependent output map or dependent output column is created;
When the assignment symbol := is used, the dependency link is immediately broken after the output map/column has been calculated

MapCalc and TabCalc:
The same type of formulas can be used on columns in tables (called Table Calculation or TabCalc) and also on maps in the command line to the main window (called Map Calculation or MapCalc).

Undefined values:
These are indicated by a question mark (?). They either indicate missing values, unknown values, values outside the value range, or the area outside the study area.

So far you have only been looking at the content of the maps. You will now start with the actual analysis. A statistical analysis should be done using landslides with same characteristics. That is why we will separate the fossil landslides from the recent ones. We do that using a map calculation formula.

☞

- Open the table **Landslide_ID**.
- We are going to use only the class **S** (scarp) and the activities A (=Active) and R(=Reactivated). We make now a column in the table in which these will have a value 1 and the rest a value of 0. Type the following formula on the command line in the table:

Active1:=iff(((Activity="a")or(Activity="r"))and(Part="s"),1,0)

- Meaning: if the columns **Activity** has the class a (active) or r (reactivated) and the column **Part** is s (scarp), then the result is 1 otherwise 0. How many landslide fulfill this criteria?
- Close the table. We will now make an attribute map. Select *Operations / Raster Operations / Attribute map*. Select the Raster map: **Landslide_ID**, Table: **Landslide_ID**, Attribute: **Active1**. Name the output map as: **Active1.** Check the resulting map.
- We still need to change the undefined values in the map to 0 values. Type the following command line in the Main Window:

  Active:=iff(isundef(Active1),0,Active1)
- Meaning: iff the map Active1 is undefined, then we change it into 0 , otherwise we keep the same values.

Note: if you are not interested in learning the exact procedure on how the information value method can be calculated, you can simply skip this part of the exercise and go to the part dealing with the use of a script which will automate this procedure.

Step 1: Crossing the parameter maps with the landslide map

Map crossing:
The Cross operation performs an overlay of two raster maps. Pixels on the same positions in both maps are compared; the occurring combinations of class names, identifiers or values of pixels in the first input map and those of pixels in the second input map are stored. These combinations give an output cross map and a cross table. The cross table includes the combinations of input values, classes or IDs, the number of pixels that occur for each combination and the area for each combination.

The landslide occurrence map, showing only the recent landslides (Active) can be crossed with the parameter maps. In this case the map **Slope_cl** is selected as example. First the map crossings between the occurrence map and the two parameter maps have to be carried out.

☞

- Select from the main ILWIS menu the options: *Operations, Raster operations, Cross*.
- Select the map **Slope_cl** as the first map, the map **Active** as the second map, and call the output table **Actslope**. (Ignoring the undefined values has no effects, as both maps don't have undefined values). Deselect the box Output map.Click Show and OK. Now the crossing of the two maps takes place.
- Have a look at the resulting cross table. As you can see this table contains the combinations of the classes from the map **Slope_cl** and the two types from the map **Active**. Close the table.

Now the amount of pixels with different landslide activities in each slope class, has been calculated, the landslide densities can be calculated.

Step 2: Calculating landslide densities

After crossing the maps, the next step is to calculate density values. The cross-table includes the columns that will be calculated during this exercise. Each of the calculation steps is indicated below.

☞

- Make sure that the cross-table **Actslope** is opened.

  Step 2.1:In this table create a column in which only the active landslide are indicated by typing the following formula on the command line of the table window:

  AreaAct=iff(Active=1,area,0)↵

  You do this in order to calculate for each slope class the area with only active landslides.
- **Step 2.2:** Calculate the total area in each slope class.
  Select from the table menu: Columns, Aggregation.
  Select the column: **Area**. Select the function Sum. Select group by column **Slope_cl**. Deselect the box Output Table, and enter the output column **Areasloptot**. Press OK. Select a precision of 1.0.
- **Step 2.3**: Calculate the area with active landslides in each slope class.
  Again select from the table menu: Column, Aggregation.
  Select the column: **AreaAct**, Select the function Sum, select Group by column **Slope_cl**. Deselect the box Output Table, and enter the output column: **Areaslopeact**. Press OK. Select a precision of 1.0.
- **Step 2.4:** calculate the total area in the map.
  Again select from the table menu: Columns, Aggregation.
  Select the column: **Area**. Select the function Sum. Deselect the box group by. Deselect the box Output table, and enter the output column: **Areamaptot**. Press OK. Select a precision of 1.0.
- **Step 2.5:** The next step is to calculate the total area with landslides in the map. Again select from the table menu: Columns, Aggregation.
  Select the column: **AreaAct**. Select the function Sum. Deselect the box group by. Deselect the box Output Table, and enter the output column: **Areamapact**. Press OK. Select a precision of 1.0.
- **Step 2.6:** Calculate the landslide density per slope class
  Type:

  Densclas=Areaslopeact/Areasloptot↵

  Select a precision of 0.0001.
- **Step 2.7:** Calculate the landslide density for the entire map.
  Type:

  Densmap=Areamapact/Areamaptot↵

  Select a precision of 0.0001 and decimal: 4.

Hint: If Denclas and Densmap are not in 4 decimals then use property dialog box to change the decimal

The result will look like below:

	Slope_cl	Active	NPix	Area	AreaAct	Areasloptot	Areaslopeact	Areamaptot	Areamapact	Densclas	Densmap
0 - 5 * 0	0 - 5	0	4169438	4169438	0	4173424	3986	14000000	213446	0.0010	0.0152
0 - 5 * 1	0 - 5	1	3986	3986	3986	4173424	3986	14000000	213446	0.0010	0.0152
5 - 10 * 0	5 - 10	0	2718437	2718437	0	2723958	5521	14000000	213446	0.0020	0.0152
5 - 10 * 1	5 - 10	1	5521	5521	5521	2723958	5521	14000000	213446	0.0020	0.0152
10 - 15 * 0	10 - 15	0	1941860	1941860	0	1952714	10854	14000000	213446	0.0056	0.0152
10 - 15 * 1	10 - 15	1	10854	10854	10854	1952714	10854	14000000	213446	0.0056	0.0152
15 - 20 * 0	15 - 20	0	1488289	1488289	0	1502075	13786	14000000	213446	0.0092	0.0152
15 - 20 * 1	15 - 20	1	13786	13786	13786	1502075	13786	14000000	213446	0.0092	0.0152
20 - 25 * 0	20 - 25	0	1062314	1062314	0	1086549	24235	14000000	213446	0.0223	0.0152
20 - 25 * 1	20 - 25	1	24235	24235	24235	1086549	24235	14000000	213446	0.0223	0.0152
25 - 30 * 0	25 - 30	0	826051	826051	0	854335	28284	14000000	213446	0.0331	0.0152
25 - 30 * 1	25 - 30	1	28284	28284	28284	854335	28284	14000000	213446	0.0331	0.0152
40 - 50 * 0	40 - 50	0	407252	407252	0	450340	43088	14000000	213446	0.0957	0.0152
40 - 50 * 1	40 - 50	1	43088	43088	43088	450340	43088	14000000	213446	0.0957	0.0152
30 - 40 * 0	30 - 40	0	1017888	1017888	0	1073296	55408	14000000	213446	0.0516	0.0152
30 - 40 * 1	30 - 40	1	55408	55408	55408	1073296	55408	14000000	213446	0.0516	0.0152
50 - 60 * 0	50 - 60	0	125097	125097	0	147443	22346	14000000	213446	0.1516	0.0152
50 - 60 * 1	50 - 60	1	22346	22346	22346	147443	22346	14000000	213446	0.1516	0.0152
60 - 90 * 0	60 - 90	0	29928	29928	0	35866	5938	14000000	213446	0.1656	0.0152
60 - 90 * 1	60 - 90	1	5938	5938	5938	35866	5938	14000000	213446	0.1656	0.0152

Now you have calculated all the required densities for the map **Slope._cl**

Step 3: Calculating weight values

The final weight-values are calculated by taking the natural logarithm of the density in the class, divided by the density in the map. With this calculation we find that the density in the entire map = 213446 / 14000000 = 0.0152

Previously the calculation was done on the cross-table for the maps **Slope_cl** and **Active**. As you could see from the table above, this results in many redundant values, since you only want to calculate the densities and the weights for each slope class. The result should look like table below instead, where each slope class occupies only one record. That is why you will work now with the attribute table connected to the map **Slopecl** and use table joining combined with aggregation to obtain the data from the cross table.

	Areasloptot	Areaslopact	Densclas	Weight
0 - 5	4173424	3986	0.0010	-2.7213
5 - 10	2723958	5521	0.0020	-2.0281
10 - 15	1952714	10854	0.0056	-0.9985
15 - 20	1502075	13786	0.0092	-0.5021
20 - 25	1086549	24235	0.0223	0.3833
25 - 30	854335	28284	0.0331	0.7782
30 - 40	1073296	55408	0.0516	1.2222
40 - 50	450340	43088	0.0957	1.8399
50 - 60	147443	22346	0.1516	2.3000
60 - 90	35866	5938	0.1656	2.3883

☞

- Create a table **Slope_cl** for the domain **Slope_cl**. This table contains no additional columns, except the column with the domain. Repeat the procedure from above, but now with table joining.
- **Step 1:** Calculate the total area in each slope class.
  Select Columns, Join. Select table **Actslope**. Select column: Area. Select function Sum. Select group by column: **Slopecl**. Select output column **Areasloptot**. Press OK.
- **Step 2:** Calculate the area with active landslides in each slope class.
  Select Columns, Join. Select table: **Actslope**. Select column **Areaact**. Select function Sum. Select group by column **Slopecl**. Select output column **Areaslopact**. Press OK.
- **Step 3:** With both columns, you can calculate the landslide density in each slope class with the formula:

  Densclas:=Areaslopact/Areasloptot↵

  Select a precision of 0.0001.
- If you look at the result, some classes have a density of 0. This should be

adjusted, since the calculation of the weights is not possible. To adjust type the following formula:

Dclas:=iff(Densclas=0,0.0001,Densclas)↵

- The final weight can now be calculated with the formula:

  Weight:=ln(Dclas/0.0152)↵

- Check the resulting weights in the table. Which slope classes have the most important relation with landslides?
- Close the table.

Step 4: Creating the weight maps

The weights from the table can now be used to renumber the maps.

☞

- Select from the main ILWIS menu: Operations, Raster operations, Attribute map. Select raster map **Slope_cl,** table **Slope_cl**. Select attribute **Weight**. Select output raster map **Wslope_cl**. Press OK.
- Display the resulting map **Wslope_cl**. Stretch between -2.5 and +2.5
- Use the same procedure the other parameter map **Lithology**. Name table as **Lithology_cl** with domain **lithology**. The resulting map should be called: **WLithology**.
- The weights for the two maps can be added with the formula:

  Weight1=Wslope_cl+WLithology↵

- Display the map **Weight1** and use the pixel information window in order to read the information from the maps **Slope_cl, Wslopecl, Lithology, WLithology** and **Weight1**.

Step 5: Use of scripts (to calculate for all factor maps)

You can automate the calculation procedure by using a script, which contains the formulas for the ILWIS operations. Parameters can be used in the form of %1 - %9. You can make a script by copying the statement which is shown on the command line when executing an operation, and pasting it into a script file. Table calculation formulas need the word TABCALC in front. For more information on scripts, consult the ILWIS Help, or the ILWIS User's Guide

Script:
A script is a sequenced list of ILWIS commands and expressions. By creating a script, you can build a complete GIS or Remote Sensing analysis for your own research discipline. Each line in a script is a statement that is executed via the ILWIS command line of the Main window. Via a script, you can for instance handle some necessary object management (e.g. copy or delete), display of objects (open or show), and the creation and calculation of data objects. All map and table calculations, and all ILWIS expressions to perform operations may be used. Furthermore, you can call other scripts and start other Windows applications from within a script.

☞

- The script that is given on the next page can be used to automate the analysis. Select *File/Create/Script*, and copy the text in the script window. Save the script as **Weights**
- Then close the script and run the script on the command line:

  Run weights Slope_cl

- Similarly you can also run the script for other parameter maps that you consider important for landslide occurrence. Such as Lithology, or landuse or distance from the river etc.

  Run weights aspect_cl
  Run weights Landuse
  Run weight River_dis
  Run weight Lithology
  Etc..

Parameters in scripts: A script can use parameters. Parameters in a script replace (parts of) object names, operations, etc. Parameters in scripts work as DOS replaceable parameters in DOS batch files, and must be written on the Script Tab in the script editor as %1, %2, %3, up to %9.

```
//script for Information value method
// required parameters: %1 = name of the factor map, which should be a class
map

del active%1.* -force
del %1w.* -force

//calculation in cross table
Active%1.tbt := TableCross(%1,active,IgnoreUndefs)
Calc Active%1.tbt

//Calculate the area of landslides in the crosstable only for the combinations with
landslides
Tabcalc Active%1 AAct:=iff(active=1,Area,0)

//create an attribute table
crtbl %1w %1

//calculate the total area of landslides within each class of the factor map
Tabcalc %1w Areaclassact:= ColumnJoinSum(Active%1.tbt,AAct,%1,1)

//calculate the total area of the class of the factor map
Tabcalc %1w Areaclasstot:= ColumnJoinSum(Active%1.tbt,Area,%1,1)

//calculate the total area of landslides in the map
Tabcalc %1w Areaslidetot:= ColumnJoinSum(Active%1.tbt,AAct,,1)

//calculate the total area of the map
Tabcalc %1w Areamaptot:= ColumnJoinSum(Active%1.tbt,Area,,1)

//calculate the density of landslides in the class
Tabcalc %1w dclass { vr=::0.000001}:=Areaclassact/Areaclasstot

//correcting for those areas that have no landslides
Tabcalc %1w densclass { vr=::0.000001}:= iff((isundef(dclass))or(dclass=0),
0.000001, dclass)

//calculate the density of landslides in the map
Tabcalc %1w densmap { vr=::0.000001}:=Areaslidetot/Areamaptot

//calculate the weight
Tabcalc %1w weight:=ln(densclass/densmap)

//generating the weight map
active%1:= MapAttribute(%1,%1w.tbt.weight)

Show active%1.mpr
```

After running the script you can check the weights in the attribute table to evaluate whether the parameter map is a useful tool for landslide prediction. You might also have to combine different parameters, into new more meaningful ones. This is an iterative process.

Step 6: Combining the weights in a final susceptibility map

After running the script for all the factor maps, and after selecting which maps you want to use in the creation of the final map, you can add up the weights into a final weight map.

☞

- In the command line calculate the following equation to add up the weight maps:

Weight:=activeaspect_cl+activeslope_cl+activelihtology+activelanduse+activeriver_dis

The map Weight has many values, and cannot be presented as it is as a qualitative hazard (susceptibility) map. In order to do so we first need to classify this map in a small number of units.

☞

- Calculate the histogram of the map **Weight** and select the boundary values for three classes: Low hazard, Moderate hazard, and High hazard.
- Create a new domain: **Susceptibility**. By selecting: File, Create, Create domain. The domain should be a Class and tick on Group. Now enter the names and the boundary values of the different classes in the domain. When you are ready, close the domain.
- The last step is using the program slicing. Select: Operations, Image processing, slicing. Select raster map: **Weight**. Select output raster map: hazard. Select domain: **Susceptibility**. Press show and OK.
- Evaluate the output map with Pixel information. If necessary adjust the boundary values of the domain hazard and run slicing again, until you are satisfied with the result

☞ **For experienced ILWIS users:**

- It is also important to include the areas occupied by old landslides in the hazard map. You can do this with a map calculation formula. Design the procedure and formula's yourself. Give it the name **Final**

Step 7: Calculating success rate.

The "predictive power" of the resulting weight maps can be tested by analysing their success rate and prediction rate. The success rate is calculated by ordering the pixels of a susceptibility map in a number of classes, from high to low values, based on the frequency information from the histogram. After that an overlay is made with the landslide inventory map, and the joint frequency is calculated. The success rate indicates how much percentage of all landslides occurs in the pixels with the highest values in the different combination maps. For example, 50 percent of all landslides are predicted by 10 percent of the pixels with the highest value in the map.

☞

- Create a script for the calculation of the success rate, using the example script below. Call it: **success**
- Run the success rate script as follows:

  Run success weight

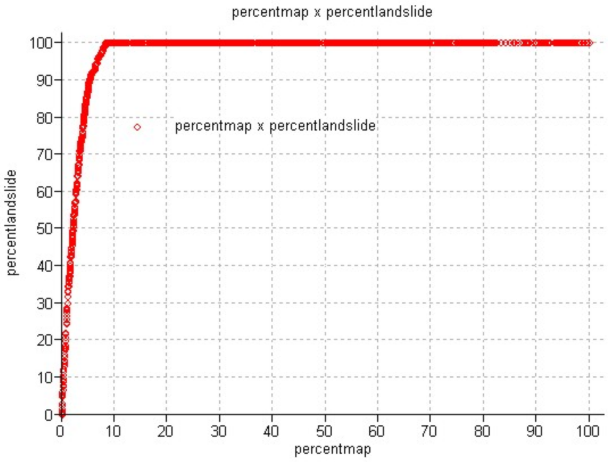
- After running the script, open the table **Activeweight**. Open the Display in a graph the column **percentmap** on the X-axis and the column **percentlandslide** on the Y-axis. Evaluate the results and decide on the best boundary values for dividing the map in high, moderate and low susceptibility.
- Use these boundaries to classify the weight map again.

```
//script for success rate calculation
// one parameter %1 = weight map resulting from the statistical analysis

del active%1.* -force

// Cross Final with Map: active
Active%1.tbt := TableCross(%1,active,IgnoreUndefs)

//In the cross table, calculate
tabcalc Active%1 npixact:=iff(active=1,npix,0)
tabcalc Active%1 Npcumactive = ColumnCumulative(npixact)
tabcalc Active%1 totalslide = ColumnAggregateSum(npixact,,1)
tabcalc Active%1 totalarea = ColumnAggregateSum(npix,,1)
tabcalc Active%1 percentage:=100*(Npcumactive / totalslide)
tabcalc Active%1 Percentlandslide:=100-percentage
tabcalc Active%1 Npixcumul:= cum(NPix)
tabcalc Active%1 reverse = totalarea -npixcumul
tabcalc Active%1 percentmap = 100*(reverse/totalarea)
// after this display a graph with Percentlandslide az y-axis and Percentmap as x-axis
```



Some final remarks:

- The method was only done using a limited number of parameter maps, just to show the procedure. In reality many more parameter are used. The method is also used to differentiate the parameters according to their importance.
- The analysis should actually be done for different landslide types separately, as they will all have different combinations of causal factors.
- The Hazard index method is a useful, but simple method. Many more methods exist for landslide hazard assessment, which might be more appropriate, given the objectives of the study, the size of the area, and the available input data.

☞ **For experienced ILWIS users:**

- There is also another script in the directory which can be used for calculating a more complicated method: Weights of Evidence. You can try that as well if you like.

```
rem ILWIS Script for Weights of Evidence
//The parameter %1 refers to the name of the factor map. It should be less than 7 characters long.
// Make sure that each map has a domain with the same name

//FIRST WE WILL DELETE EXISTING RESULT FILES
// the crosstable s%1.tbt
//The attribute table %1.tbt
// and we make a new attribute table

del s%1.*
del w%1.*
del %1.tbt
crtbl %1 %1

//NOW WE CROSS THE FACTOR MAP WITH THE ACTIVITY MAP
// The landslide map should be called ACTIVE and should have either 0 or 1 values. 1 values mean
landslides.
// The cross table is called s%1

s%1=TableCross(%1.mpr,active.mpr,IgnoreUndefs)
calc s%1.tbt

//Now we calculate one column in the cross table to indicate only the pixels with landslides.

Tabcalc s%1 npixact=iff(active=1,NPix,0)

//NOW WE USE AGGREGATION FUNCTION, WITH OR WITHOUT A KEY TO CALCULATE:
//NCLASS = number of pixels in the class. We sum the values from columns Npix and group them by %1
//nslclass = number of pixels with landslides in the class.We sum the values from columns Npixact and
group them by %1
//nmap = number of pixels with landslides in the map. We sum the values from columns Npix and don't
group them
//nslide = number of pixels with landslide in the map. We sum the values from columns Npixact and don't
group them
//THE RESULTS ARE NOT STORED IN THE CROSS TABLE S%1 BUT IN THE ATTRIBUTE TABLE %1

Tabcalc s%1 %1.nclass = ColumnJoinSum(s%1.tbt,Npix,%1,1)
Tabcalc s%1 %1.nslclass = ColumnJoinSum(s%1.tbt,Npixact,%1,1)
Tabcalc s%1 %1.nmap = ColumnJoinSum(s%1.tbt,Npix,,1)
Tabcalc s%1 %1.nslide = ColumnJoinSum(s%1.tbt,Npixact,,1)

//NOW WE CALCULATE THE FOUR VALUES NPIX1 - NPIX4 AS INDICATED IN THE EXERCISE BOOK. THIS IS
DONE IN THE ATTRIBUTE TABLE
// We correct for the situation when Npix1 - Npix3 might be 0 pixels, and change it into 1 pixel

Tabcalc %1 npix1 =IFF((nslclass>0),nslclass,1)
Tabcalc %1 npix2 = IFF((nslide-nslclass)=0,1,nslide-nslclass)
Tabcalc %1 npix3 = IFF((nclass-nslclass)=0,1,nclass-nslclass)
Tabcalc %1 npix4 = nmap-nslide-nclass+nslclass

//NOW WE CALCULATE THE WEIGHTS IN THE ATTRIBUTE TABLE
Tabcalc %1 wplus {dom=value.dom; vr=-10:10:0.00001} =
LN((npix1/(npix1+npix2))/(npix3/(npix3+npix4)))
Tabcalc %1 wminus {dom=value.dom; vr=-10:10:0.000001}  =
LN((npix2/(npix1+npix2))/(npix4/(npix3+npix4)))

//NOW WE CALCULATE THE CONTRAST FACTOR
Tabcalc %1 Cw = wplus-wminus

//NOW WE CALCULATE THE FINAL WEIGHT
//The final weight is the sum of the positive weight and the negative weights of the other classes
Tabcalc %1 WminSum=aggsum(wminus)
Tabcalc %1 Wmap=wplus+Wminsum-Wminus

//NOW WE MAKE AN ATTRIBUTE MAP OF THE FINAL WEIGHTS
w%1.mpr = MapAttribute(%1,%1.Wmap)
calc w%1.mpr
```