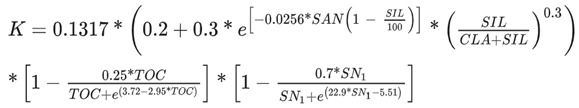
**RUSLE**

R Factor

Downloaded from ESDAC - European Soil Data Centre

K Factor



Where, K is soil erodibility factor (t ha h ha−1 MJ−1 mm−1), SAN is sand weight content (%), SIL is silt weight content (%), CLA is clay weight content (%), TOC is soil organic carbon content (%), SN1=1 – (SAN / 100).

**Code GEE** :

//kheda district shap file

var table = ee.FeatureCollection("projects/ee-sjatin/assets/Kheda\_District");

var c1 = ee.Image("projects/ee-sjatin/assets/carbon0\_5") // add carbon(0-5) file path

var c2 = ee.Image('projects/ee-sjatin/assets/carbon5\_15') // add carbon(5-15) file path

var c3 = ee.Image("projects/ee-sjatin/assets/carbon15\_30") // add carbon(15-30) file path

var carbon = ee.Image([c1, c2, c3]).rename(['c1', 'c2', 'c3'])

// carbon layer average(0-5,5-15,15-30)

var factorc = carbon.expression(

'((c1+c2+c3)/3)',

{'c1': carbon.select('c1'), 'c2': carbon.select('c2'), 'c3': carbon.select('c3') });

var s1 = ee.Image("projects/ee-sjatin/assets/sand0\_5") // add sand(0-5) file path

var s2 = ee.Image('projects/ee-sjatin/assets/sand5\_15') // add sand(5-15) file path

var s3 = ee.Image("projects/ee-sjatin/assets/sand15\_30") // add sand(15-30) file path

var sand = ee.Image([s1, s2, s3]).rename(['s1', 's2', 's3'])

// sand layer average(0-5,5-15,15-30)

var factors = sand.expression(

'((s1+s2+s3)/3)',

{'s1': sand.select('s1'), 's2': sand.select('s2'), 's3': sand.select('s3') });

var si1 = ee.Image("projects/ee-sjatin/assets/silt0\_5") // add silt(0-5) file path

var si2 = ee.Image('projects/ee-sjatin/assets/silt5\_15') // add silt(5-15) file path

var si3 = ee.Image("projects/ee-sjatin/assets/silt15\_30") // add silt(15-30) file path

var silt = ee.Image([si1, s2, s3]).rename(['si1', 'si2', 'si3'])

// silt layer average(0-5,5-15,15-30)

var factorsi = silt.expression(

'((si1+si2+si3)/3)',

{'si1': silt.select('si1'), 'si2': silt.select('si2'), 'si3': silt.select('si3') });

var cl1 = ee.Image("projects/ee-sjatin/assets/clay0\_5") // add clay(0-5) file path

var cl2 = ee.Image('projects/ee-sjatin/assets/clay5\_15') // add clay(5-15) file path

var cl3 = ee.Image("projects/ee-sjatin/assets/clay15\_30") // add clay(15-30) file path

var clay = ee.Image([cl1, cl2, cl3]).rename(['cl1', 'cl2', 'cl3'])

// clay layer average(0-5,5-15,15-30)

var factorcl = clay.expression(

'((cl1+cl2+cl3)/3)',

{'cl1': clay.select('cl1'), 'cl2': clay.select('cl2'), 'cl3': clay.select('cl3') });

var sand = factors.divide(10) //convert gram per kg file into percentage

var silt = factorsi.divide(10) //convert gram per kg file into percentage

var clay = factorcl.divide(10) //convert gram per kg file into percentage

var orgcar = factorc.divide(100) //convert decigram per kg file into percentage

var soil = ee.Image([sand, silt, clay, orgcar]).rename(['sand', 'silt', 'clay', 'orgcar'])

// calculate k factor

var factorK = soil.expression(

'(0.1317\*(0.2+0.3 \* exp(-0.0256 \* SAND \* (1 - (SILT/100))))((SILT/(SILT + CLAY))0.3) (1 - (0.25 \* CORG/(CORG + exp(3.72 - 2.95 \* CORG)))) \* (1 - (0.7 \* (1-(SAND/100))/((1-(SAND/100))+ exp(-5.51 +22.9 \* (1-(SAND/100)))))))',

{'SAND': soil.select('sand'), 'SILT': soil.select('silt'), 'CLAY': soil.select('clay'), 'CORG': soil.select('orgcar') });

Map.addLayer(factorK.clip(table))//add layer into map

// export file into drive

Export.image.toDrive({

image: factorK,

description: 'factorK\_export', // Name of the exported file

folder: 'factorK\_folder', // Name of the folder in Google Drive

region: factorK.geometry(), // Region to export

scale: 250, // Spatial resolution in meters

maxPixels: 1e13 // Maximum number of pixels to export

});

LS Factor

**Use Arcgis**

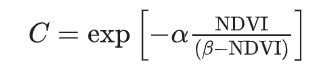
DEM file > Flow Direction > Flow Accumulation > Slope

Formula:

Power("FlowAcc"\*cell\_size/22.13,0.4)\*Power(Sin("Slope"\*0.01745)/0.0896,1.4)\*1.4

where flow acc. is the accumulated slope effect on soil erosion, map resolution is the grid size of the map, and sin slope is the slope degree of land in sin.

C Factor



where α and β are unitless parameters that determine the shape of the curve relating to NDVI and the C-factor.

**Code GEE :**

// add ndvi file path

var ndvi\_median = ee.Image("projects/ee-sjatin/assets/NDVI\_Data1")

// calculate c factor

var factorC = ndvi\_median.expression(

'(exp(-2 \*(( NDVI)/(1 - NDVI))))', {

'NDVI': ndvi\_median, });

factorC = factorC.clamp(0, 0.9);//clamps the values in all bands of an image to all lie within the specified range

Map.addLayer(factorC)//add layer into map

// export file into drive

Export.image.toDrive({

image: factorC,

description: 'factorC', // Name of the exported file

folder: 'factorC', // Name of the folder in Google Drive

region: factorC.geometry(), // Region to export

scale: 10, // Spatial resolution in meters

maxPixels: 1e13 // Maximum number of pixels to export

});

P Factor

**Use Arcgis**

1. DEM file
2. LULC file

Calculate slope > Reclassify > Assign values in attribute table

**Calculate Taluka Wise Mean**

var soil\_loss = ee.Image(R.multiply(K).multiply(LS).multiply(C).multiply(P)).rename("Soil Loss")

var roi = ee.FeatureCollection("add path")

var style = ['white','lightgreen','black','orange','darkred','white']

Map.addLayer (soil\_loss, {min: 0, max: 10, palette: style}, 'Soil Loss',0)

var SL\_class = soil\_loss.expression(

"(b('Soil Loss') < 5) ? 1" +

": (b('Soil Loss') < 10) ? 2" +

": (b('Soil Loss') < 15) ? 3"+

": (b('Soil Loss') < 20) ? 4"+

": (b('Soil Loss') < 40) ? 5"+

": 6")

.rename('SL\_class').clip(aoi);

Map.addLayer (SL\_class, {min: 0, max: 6, palette: style}, 'Soil Loss Class')

var SL\_mean = soil\_loss.reduceRegion({

geometry: aoi,

reducer: ee.Reducer.mean(),

scale: 30,

maxPixels: 475160679

})

print ("Mean Soil Loss",SL\_mean.get("Soil Loss"))

// Load the talukas shapefile as a feature collection

var talukas = ee.FeatureCollection('users/deepp6603/Taluka'); // Replace 'path/to/talukas/shapefile' with the correct path to the talukas shapefile

// Define the name of the subdistrict you want to focus on

var targetSubdistrictName = 'Thasra'; // Replace 'YourSubdistrictName' with the name of the subdistrict you want to focus on

// Filter the talukas shapefile to find the target subdistrict

var targetSubdistrict = talukas.filter(ee.Filter.eq('subdistric', targetSubdistrictName)).first();

// Check if the target subdistrict exists

if (targetSubdistrict) {

// Calculate the mean soil loss for the target subdistrict

var meanSoilLoss = soil\_loss.reduceRegion({

geometry: targetSubdistrict.geometry(),

reducer: ee.Reducer.mean(),

scale: 30,

maxPixels: 475160679

}).get('Soil Loss');

// Print the mean soil loss for the target subdistrict

print('Mean Soil Loss for Subdistrict', targetSubdistrictName, meanSoilLoss);

} else {

// Print a message if the target subdistrict is not found

print('Subdistrict', targetSubdistrictName, 'not found in the shapefile');

}

**Data Download**

**NDVI data :**

var s2 = ee.ImageCollection("COPERNICUS/S2\_SR\_HARMONIZED")

var roi = ee.FeatureCollection("add path")

var date1='2017-01-01'

var date2='2023-01-01'

s2 = s2.filterDate(date1, date2).median().clip(roi);

var image\_ndvi = s2.normalizedDifference(['B8','B4']).rename("NDVI");

Map.addLayer(image\_ndvi, {min: 0.059, max: 0.46}, 'NDVI');

//4..NDVI

Export.image.toDrive({

image:image\_ndvi.clip(roi),

description: 'NDVI\_Data',

scale: 10,

region: roi,

maxPixels: 10000000000000,

folder: 'Rusle\_Project/Input' // Replace with the actual path to your folder

});

**DEM data :**

var dataset = ee.Image('USGS/SRTMGL1\_003');

var roi=ee.FeatureCollection("add path")

// Export the slope image to Google Drive

Export.image.toDrive({

image: dataset.clip(roi),

description: 'dem\_data',

scale: 30,

region: roi,

maxPixels: 1e13,

folder: 'Rusle\_Project/Input' // Replace with the actual path to your folder

});