EUROPEAN COMMISSION
EUROSTAT

Directorate E: Sectoral and regional statistics
Unit E-4: Regional statistics and geographical information

Methodological notes for the calculation of Total Surface Area and Land surface area for all NUTS

Document History

Version	Date	Comment	Modified Pages
1.00	15.11.2017	Version for NUTS 2013	

Contents

1. Methodological notes

The calculation is done in ArcGIS using the script below. It calculates for each NUTS level (e.g. 0..3) from the GISCO NUTS dataset at scale 1:100 000 the total surface area (TSA) in square meters in ETRS1989 Lambert Equal Area Projection and exports these into a dbf table. The same procedure was applied to the GISCO Country dataset. For the calculation of LSA the classified satellite image from the CORINE 2006 was processed. Due to missing CORINE data for Greece data from the year 2000 from CORINE 2000 were used. For each NUTS region, the sum of water pixels reclassified into a binary image was calculated and subtracted from TSA. All results were imported into an MS-ACCESS Database for further aggregation.

2. Processing Script

```
# total area calculation
# country
import arcpy,os,numpy
import utils, datetime


from arcpy import env

# Set overwrite option
arcpy.env.overwriteOutput = True


def unique_values_np(table, field):
    data = arcpy.da.TableToNumPyArray(table, [field])
    return numpy.unique(data[field])
def dropfield(dataset,fieldlist): # does a field drop one by one therefore not
choking if a defined field is not avaiable.
    desc = arcpy.Describe(dataset)
    fieldnames =[]
    for x in desc.fields:
        fieldnames.append(x.name)
    for x in fieldlist:
                if x in fieldnames:
                    arcpy.DeleteField_management(dataset, [x])
    return()

# define data sources
nutsregions = r'\\s-tpol\connection-files\PROD\PROD-SDE-
GISCOVIEW.sde\GISCOVIEW.NUTS_RG_100K_2010_LAEA'
cntryregions = r'\\s-tpol\connection-files\PROD\PROD-SDE-
GISCOVIEW.sde\GISCOVIEW.CNTR_RG_100K_2010_LAEA'
CORINE = r'\\s-tlar\development\CLC\CLC.gdb\clc06_27052013_D'  #
clc06_27052013_D_1
CORINE = r'\\s-tpol\CORINE\g100_06'
# get data from ESTAT pop stats for comparison
ESTATDATAURL_NUTS3=
"http://ec.europa.eu/eurostat/data/database?node_code=tgs00002"
```

```
ESTATDATAURL_NUTS3= "http://ec.europa.eu/eurostat/estat-navtree-portlet-
prod/BulkDownloadListing?file=data/demo_r_d3area.tsv.gz"
# quickfix for import # remove ":"; fix dec import in xls manually

# ### demo_r_d3area
workdir = r'C:/temp/work/'
workdir = r'J:\develop\JIRA-2015\GISCO-359 (Total surface area)'
env.workspace= workdir
# # start processing
print ('start time:'+str(datetime.datetime.now()))
proxy = urllib2.ProxyHandler({'http': removed for security purpose'})
auth = urllib2.HTTPBasicAuthHandler()
opener = urllib2.build_opener(proxy, auth, urllib2.HTTPHandler)
urllib2.install_opener(opener)
getfile(ESTATDATAURL, workdir+os.sep+'tgs.zip') # download file
unzip(workdir+os.sep+'tgs.zip', workdir+os.sep+'tgs.tsv') # unzip it
# # import into DB
# # make watermask from CORINE
#
# # fix nuts
arcpy.MakeFeatureLayer_management(nutsregions, "nutslyr")
arcpy.CopyFeatures_management("nutslyr",workdir+os.sep+"nutslyrfull.shp") #
make local shp copy
if arcpy.Exists("nutslyr"):
      arcpy.Delete_management("nutslyr")
arcpy.MakeFeatureLayer_management(workdir+os.sep+"nutslyrfull.shp", "nutslyr")
fieldlist=
("CNTR_CODE","CNTR_NAME","CNTR_NAME_","CNTR_NAME1","CNTR_NAME_1","CNTR_NAME_2"
,"CNTR_ISO3_","NUTS_NAME_","NUTS_NAME1","NUTS_NAM_1")
# drop unnessary fields
dropfield("nutslyr",("CNTR_CODE","CNTR_NAME","CNTR_NAME_","CNTR_NAME1","CNTR_N
AME_1","CNTR_NAME_2","CNTR_ISO3_","NUTS_NAME_","NUTS_NAME1","NUTS_NAM_1")) #
CNTR_CODE,CNTR_NAME,CNTR_NAME_,CNTR_NAME1,CNTR_NAME_1,CNTR_NAME_2,CNTR_ISO3_,N
UTS_NAME_,NUTS_NAME1,NUTS_NAM_1,
#
arcpy.AddField_management("nutslyr", 'cntr', "TEXT", "", "", "","",
"NULLABLE")
arcpy.CalculateField_management("nutslyr", 'cntr', '!NUTS_ID![0:2]',
'PYTHON_9.3')
arcpy.AddField_management("nutslyr", 'area', "DOUBLE", "", "", "","", "")
arcpy.CalculateField_management("nutslyr", 'area', 'float(!shape.area!)',
'PYTHON_9.3')
# # sum up to NUTS0..3 level for each country
for n in (0,1,2,3):
     arcpy.SelectLayerByAttribute_management("nutslyr", "NEW_SELECTION",
'"STAT_LEVL_" = '+str(n))
     if arcpy.Exists(workdir+os.sep+"nuts_lvl"+str(n)+'.shp'):
      arcpy.Delete_management(workdir+os.sep+"nuts_lvl"+str(n)+'.shp')
#
arcpy.CopyFeatures_management("nutslyr",workdir+os.sep+"nuts_lvl"+str(n)+'.shp
') # make local file for import into accessx
#     arcpy.Statistics_analysis("nutslyr",
workdir+os.sep+"nutslyr"+'s'+str(n)+'.dbf', [['area','SUM']])
print 'done nuts area analysis'
# # calculate Area from CNTR_LAYER
# # select only the unique NUTS lvl 0 countries
if arcpy.Exists("nutslyr"):
      arcpy.Delete_management("nutslyr")
arcpy.MakeFeatureLayer_management(workdir+os.sep+"nutslyrfull.shp", "nutslyr")
arcpy.SelectLayerByAttribute_management("nutslyr", "NEW_SELECTION",
'"STAT_LEVL_" = 0')
cntr2beselected = unique_values_np("nutslyr", 'NUTS_ID')
arcpy.MakeFeatureLayer_management(cntryregions, "cntrlyr")
fixcntrsel ="('"+ "','".join(cntr2beselected)+"')"
arcpy.SelectLayerByAttribute_management("cntrlyr", "NEW_SELECTION", '"CNTR_ID"
in '+fixcntrsel)
if arcpy.Exists(workdir+os.sep+'cntr_sel.shp'):
      arcpy.Delete_management(workdir+os.sep+'cntr_sel.shp')
arcpy.CopyFeatures_management("cntrlyr",workdir+os.sep+'cntr_sel.shp')
if arcpy.Exists("cntrlyr"):
      arcpy.Delete_management("cntrlyr")
arcpy.MakeFeatureLayer_management(workdir+os.sep+'cntr_sel.shp', "cntrlyr")
arcpy.AddField_management("cntrlyr", 'area', "DOUBLE", "", "", "","", "")
arcpy.CalculateField_management("cntrlyr", 'area', 'float(!shape.area!)',
'PYTHON_9.3')
if arcpy.Exists(workdir+os.sep+'cntr_area.shp'):
      arcpy.Delete_management(workdir+os.sep+'cntr_area.shp')
arcpy.CopyFeatures_management("cntrlyr",workdir+os.sep+'cntr_area.shp')
arcpy.Statistics_analysis("cntrlyr",
workdir+os.sep+"cntrlyr"+'areasum'+'.dbf', [['area','SUM']])
print 'done country area analysis'



### water bodies from corine landcover 2006 - latest currently avaiable
from arcpy.sa import *
arcpy.CheckOutExtension("Spatial")
# reclass ALL water bodies to 1 - rest to 0
# arcpy.gp.Reclassify_sa("g100_06","VALUE","1 39 0;40 44 1;48 NODATA;49 0;50
1;255 NODATA",workdir+os.sep+"corine_rec1","NODATA")
# zonalstats(SUM) over NUTSIDs
for n in (0,1,2,3):
    if not arcpy.Exists(workdir+os.sep+"watersumlvl"+str(n)+'.dbf'):
        try:
                if not arcpy.Exists(workdir+os.sep+"rnutslvl"+str(n)):

arcpy.PolygonToRaster_conversion(workdir+os.sep+"nuts_lvl"+str(n)+'.shp',
"NUTS_ID",
workdir+os.sep+"rnutslvl"+str(n),'','',workdir+os.sep+"corine_rec1")
                out =
ZonalStatisticsAsTable(workdir+os.sep+"rnutslvl"+str(n),"VALUE",workdir+os.sep
+"corine_rec1",workdir+os.sep+"watersumlvl"+str(n)+'.dbf',"DATA","SUM")
                print 'stats sucessfull for LVL'+str(n)
        except: #
arcpy.FeatureToRaster_conversion(in_features="nuts_lvl3",field="NUTS_ID",out_r
aster="C:/temp/work/rnutslvl3",cell_size="J:/develop/JIRA-2015/GISCO-359
(Total surface area)/corine_rec1")
            print 'conversion not sucessfull for'+str(n)
    if not arcpy.Exists(workdir+os.sep+"watersumVATlvl"+str(n)+'.dbf'):
            arcpy.MakeTableView_management(workdir+os.sep+"rnutslvl"+str(n),
"view", "", "", "")
            arcpy.CopyRows_management("view",
workdir+os.sep+"watersumVATlvl"+str(n)+'.dbf')
            print 'export VAT for lvl'+str(n)
print 'done water surface analysis'
print ('end time:'+str(datetime.datetime.now()))

# GRID_CODE    CLC_CODE    LABEL1    LABEL2    LABEL3
# 1    111    Artificial surfaces    Urban fabric    Continuous urban fabric
# 2    112    Artificial surfaces    Urban fabric    Discontinuous urban
fabric
# 3    121    Artificial surfaces    Industrial, commercial and transport
units    Industrial or commercial units
# 4    122    Artificial surfaces    Industrial, commercial and transport
units    Road and rail networks and associated land
# 5    123    Artificial surfaces    Industrial, commercial and transport
units    Port areas
# 6    124    Artificial surfaces    Industrial, commercial and transport
units    Airports
# 7    131    Artificial surfaces    Mine, dump and construction sites
Mineral extraction sites
# 8    132    Artificial surfaces    Mine, dump and construction sites    Dump
sites
# 9    133    Artificial surfaces    Mine, dump and construction sites
Construction sites
# 10    141    Artificial surfaces    Artificial, non-agricultural vegetated
areas    Green urban areas
# 11    142    Artificial surfaces    Artificial, non-agricultural vegetated
areas    Sport and leisure facilities
# 12    211    Agricultural areas    Arable land    Non-irrigated arable land
# 13    212    Agricultural areas    Arable land    Permanently irrigated land
# 14    213    Agricultural areas    Arable land    Rice fields
# 15    221    Agricultural areas    Permanent crops    Vineyards
# 16    222    Agricultural areas    Permanent crops    Fruit trees and berry
plantations
# 17    223    Agricultural areas    Permanent crops    Olive groves
# 18    231    Agricultural areas    Pastures    Pastures
# 19    241    Agricultural areas    Heterogeneous agricultural areas
Annual crops associated with permanent crops
# 20    242    Agricultural areas    Heterogeneous agricultural areas
Complex cultivation patterns
# 21    243    Agricultural areas    Heterogeneous agricultural areas    Land
principally occupied by agriculture, with significant areas of natural
vegetation
# 22    244    Agricultural areas    Heterogeneous agricultural areas    Agro-
forestry areas
# 23    311    Forest and semi natural areas    Forests    Broad-leaved forest
# 24    312    Forest and semi natural areas    Forests    Coniferous forest
# 25    313    Forest and semi natural areas    Forests    Mixed forest
# 26    321    Forest and semi natural areas    Scrub and/or herbaceous
vegetation associations    Natural grasslands
# 27    322    Forest and semi natural areas    Scrub and/or herbaceous
vegetation associations    Moors and heathland
# 28    323    Forest and semi natural areas    Scrub and/or herbaceous
vegetation associations    Sclerophyllous vegetation
# 29    324    Forest and semi natural areas    Scrub and/or herbaceous
vegetation associations    Transitional woodland-shrub
# 30    331    Forest and semi natural areas    Open spaces with little or no
vegetation    Beaches, dunes, sands
# 31    332    Forest and semi natural areas    Open spaces with little or no
vegetation    Bare rocks
# 32    333    Forest and semi natural areas    Open spaces with little or no
vegetation    Sparsely vegetated areas
# 33    334    Forest and semi natural areas    Open spaces with little or no
vegetation    Burnt areas
# 34    335    Forest and semi natural areas    Open spaces with little or no
vegetation    Glaciers and perpetual snow
```

```
# 35    411    Wetlands    Inland wetlands    Inland marshes
# 36    412    Wetlands    Inland wetlands    Peat bogs
# 37    421    Wetlands    Maritime wetlands    Salt marshes
# 38    422    Wetlands    Maritime wetlands    Salines
# 39    423    Wetlands    Maritime wetlands    Intertidal flats
# 40    511    Water bodies    Inland waters    Water courses
# 41    512    Water bodies    Inland waters    Water bodies
# 42    521    Water bodies    Marine waters    Coastal lagoons
# 43    522    Water bodies    Marine waters    Estuaries
# 44    523    Water bodies    Marine waters    Sea and ocean
# 48    999    NODATA    NODATA    NODATA
# 49    990    UNCLASSIFIED    UNCLASSIFIED LAND SURFACE    UNCLASSIFIED LAND
SURFACE
# 50    995    UNCLASSIFIED    UNCLASSIFIED WATER BODIES    UNCLASSIFIED WATER
BODIES
# 255    990    UNCLASSIFIED    UNCLASSIFIED    UNCLASSIFIED
```