



# **Urban Growth patterns around International Borders**

**Understanding how country dynamics affect urbanization**

## Table of Contents:

- Introduction
- Methodology
- Results

# Introduction

*"International migration and urbanization are key dimensions of the process of socioeconomic development. The unprecedented movement of peoples within the borders of their own countries is undoubtedly one of the greatest transformations of humankind witnessed in the 20th century. International migration, while it has received perhaps less attention, is an equally important process in many societies."*

- Migration, Urbanization, and Development: New Directions and Issues

When it comes to territorial disputes across the globe, the list is long and ever-changing. There are now more than 150 disputes under way that involve territory, mostly in Africa, Asia, and the Pacific region, but also in Europe and America. Some disputes are on the distant horizon (Antarctica), some are long-simmering (Jammu and Kashmir), and others—like Crimea—are at their boiling point.

Many internal and external policies have contributed to this increase. The factors can be categorized in the large concepts of economic development and social development. Some cities grew as a result of increased trade between the countries while others grew because of other territorial disputes.

Most border cities in the world have seen an increase in population density over the years which can be linked to the economic influence on the demographic dynamics. From the below table you can see the population growth of municipalities along the US-Mexico border from the 1930s. It has increased seven-fold as compared to the five-fold increase of the nation.

**Table 1. Population growth on the northern Mexican Border**

Year	Border states population
1930	2,054,345
1950	3,762,963
1970	7,848,169
1990	13,246,991
2000	16,642,676

*Source:* Sedesol, 2001

For the term project I will like to analyze the urban development around international borders. Given the current political and climate scenario, global migration has been on a rise and I would like to see if this increase is in around borders especially contested borders of

- US-Mexico
- India – Pakistan
- Israel – Palestine – Syria
- Bangladesh – Myanmar
- Cyprus
- Ethiopia – Somalia
- Crimea



Source : National Geographic

# Methodology

To determine growth, I will be using the following image collections. I will use a combination of GEE and ArcPro API. The datasets are:

- LANDSAT
- MODIS
- Night Lighting
- NDVI Index
- Socioeconomic Data and Applications Center (sedac) – Anthropogenic Biomes of the World series

## Step 1:

Find the borders and create a 200 km buffer from it which will act as the mask and area of interest.

## Step 2:

Use ArcPro API to reclassify and do all the map algebra on the anthropogenic biomes data.

Use GEE to analyze and work on the other data.

## Step 3:

Present the results in the order of:

1. Urban growth around the world using SEDAC data
2. Increase in urban development using Land use data
3. Decrease in vegetation data

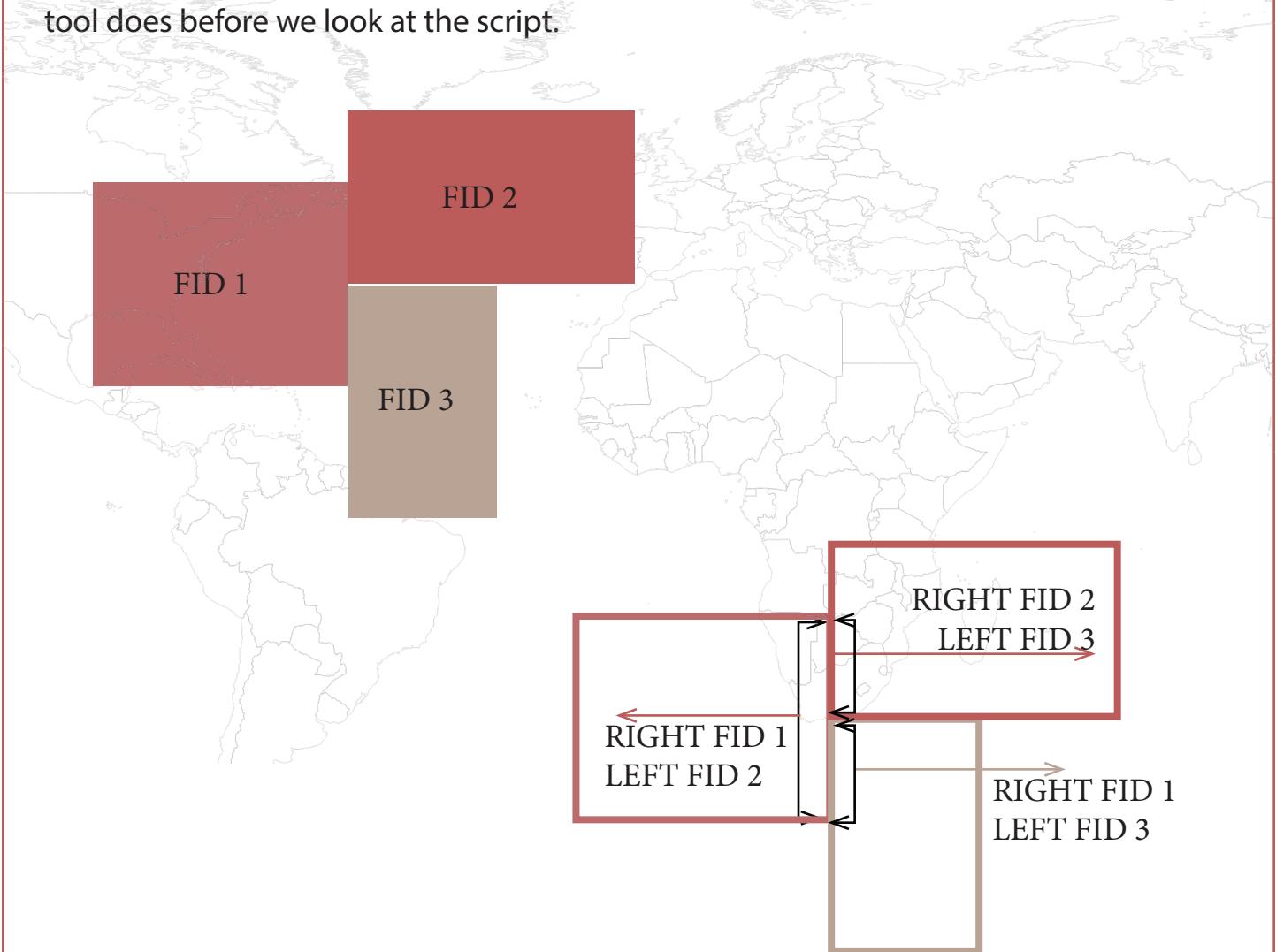
As the project using both GEE and ArcGIS Pro we will go through the steps in both individually and see how the codes and methodology differs in them.

Now lets talk in about the steps in detail and look at the scripts used at all steps.

## Step 1

**Find the borders and create a 200 km buffer from it which will act as the mask and area of interest.**

For this process I used the polygon to line feature in ArcGIS Pro. Lets visualize what the tool does before we look at the script.



As you can see from the visualization above - polygon to line create a new shapefile where each border is identified by the unique index number(FID) of the two adjacent polygons it surrounds. Using the FID I then extract the borders I am interested in and add them to the catalog as seen on the script.

# Script 1 : Polygon to line

“””

THIS SCRIPT APPLIES THE ARCGIS POLYGON TO LINE TOOL

To create an ArcGIS Pro script tool for this script, do the following.

- 1 In Catalog > Toolboxes, select an existing toolbox or create a new one.
- 2 Right-click on the entry for this toolbox in ArcToolbox, and use New > Script to open a dialog box.
- 3 In this dialog box, use Label to name the tool being created and Script File to specify its .py file.
- 4 In the same dialog box, click on Parameters and specify the following before pressing the OK button.

LABEL	DATA TYPE	DIRECTION
Input Grid	Raster Dataset	Input
Output Grid	Raster Dataset	Output

- 5 To later revise any of this, right-click to the tool's name and select Properties.

“””

```
# Import external modules
import sys, os, string, arcpy, traceback
from arcpy import env
from arcpy.sa import *

# Allow output file to overwrite any existing file of the same name
arcpy.env.overwriteOutput = True

# Check to see if Spatial Analyst license is available
if arcpy.CheckExtension("spatial") == "Available":

    try:
        # Activate ArcGIS Spatial Analyst license
        arcpy.CheckOutExtension("spatial")

        # Read user inputs from dialog box
        input_shp      = arcpy.GetParameterAsText(0)
        output_shp     = arcpy.GetParameterAsText(1)

        # Polygon to line operator
        arcpy.PolygonToLine_management(input_shp,output_shp,
                                       "IDENTIFY_NEIGHBORS")

        # Identify combinations - USA&Mexico
        arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
        arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                               "LEFT_FID = 154")
        arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                               "RIGHT_FID = 12")
        arcpy.CopyFeatures_management("MyLayerObject","USAMexico")
    
```

```

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer = (arcpy.Describe("USAMexico").file)
arcpy.MakeFeatureLayer_management("USAMexico",nameOfOutputLayer)

nameOfOutputLayerFile = "USAMexico" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Identify combinations - India&Pakistan
arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                       "LEFT_FID = 199")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                       "RIGHT_FID = 193")
arcpy.CopyFeatures_management("MyLayerObject","IndiaPakistan")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer = (arcpy.Describe("IndiaPakistan").file)
arcpy.MakeFeatureLayer_management("IndiaPakistan",nameOfOutputLayer)

nameOfOutputLayerFile = "IndiaPakistan" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Identify combinations - Isreal&Palestine
arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                       "LEFT_FID = 147")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                       "RIGHT_FID = 144")
arcpy.CopyFeatures_management("MyLayerObject","IsrealPalestine")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer = (arcpy.Describe("IsrealPalestine").file)
arcpy.MakeFeatureLayer_management("IsrealPalestine",nameOfOutputLayer)

nameOfOutputLayerFile = "IsrealPalestine" + ".lyrx"

```

```

arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Identify combinations - Isreal&Syria
arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                       "LEFT_FID = 150")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                       "RIGHT_FID = 144")
arcpy.CopyFeatures_management("MyLayerObject","IsrealSyria")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer   = (arcpy.Describe("IsrealSyria").file)
arcpy.MakeFeatureLayer_management("IsrealSyria",nameOfOutputLayer)

nameOfOutputLayerFile = "IsrealSyria" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Identify combinations - Bangladesh&Mynammar
arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                       "LEFT_FID = 232")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                       "RIGHT_FID = 228")
arcpy.CopyFeatures_management("MyLayerObject","BangladeshMynammar")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer   = (arcpy.Describe("BangladeshMynammar").file)
arcpy.MakeFeatureLayer_management("BangladeshMynammar",nameOfOutputLayer)

nameOfOutputLayerFile = "BangladeshMynammar" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

```

```

# Identify combinations - Ethopia&Somalia
 arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
 arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                         "LEFT_FID = 202")
 arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                         "RIGHT_FID = 141")
 arcpy.CopyFeatures_management("MyLayerObject","EthopiaSomalia")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer = (arcpy.Describe("EthopiaSomalia").file)
arcpy.MakeFeatureLayer_management("EthopiaSomalia",nameOfOutputLayer)

nameOfOutputLayerFile = "EthopiaSomalia" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Identify combinations - Russia&Ukraine
arcpy.MakeFeatureLayer_management(output_shp,"MyLayerObject")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "NEW_SELECTION",
                                         "LEFT_FID = 246")
arcpy.SelectLayerByAttribute_management("MyLayerObject", "SUBSET_SELECTION",
                                         "RIGHT_FID = 190")
arcpy.CopyFeatures_management("MyLayerObject","RussiaUkraine")

# Add a layer for that new shapefile to the active data frame (which is necessary because
# there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
nameOfOutputLayer = (arcpy.Describe("RussiaUkraine").file)
arcpy.MakeFeatureLayer_management("RussiaUkraine",nameOfOutputLayer)

nameOfOutputLayerFile = "RussiaUkraine" + ".lyrx"
arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")

currentProject      = arcpy.mp.ArcGISProject("CURRENT")
currentMap         = currentProject.activeMap
layerFile          = arcpy.mp.LayerFile(nameOfOutputLayerFile)
currentMap.addLayer(layerFile,"TOP")

# Deactivate ArcGIS Spatial Analyst license
arcpy.CheckInExtension("spatial")

except Exception as e:
    # If unsuccessful, end gracefully by indicating why

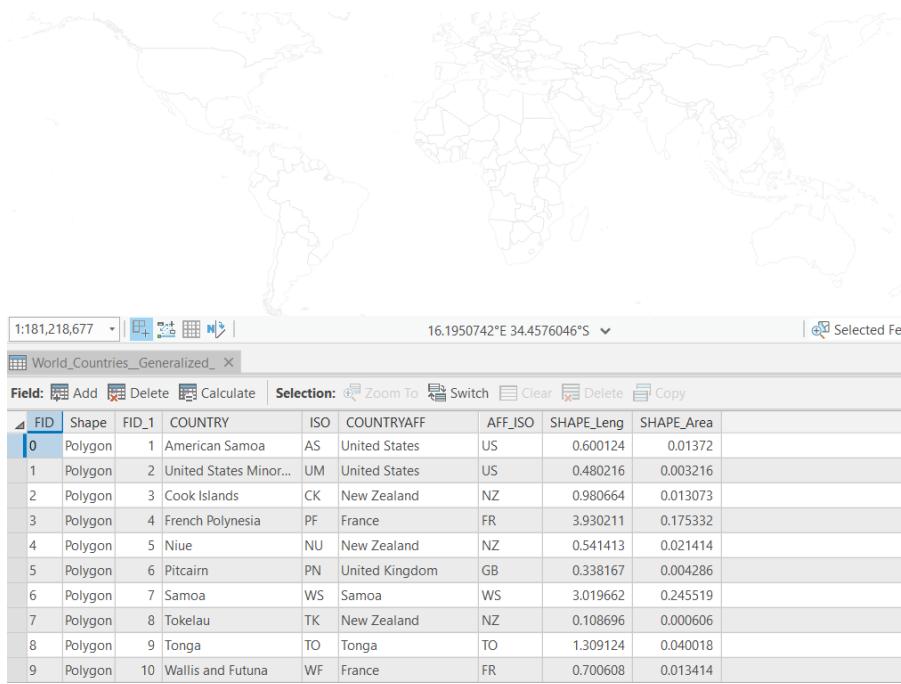
```

```

arcpy.AddError("\n" + "Script failed because: \t\t" + e.args[0] )
# ... and where
exceptionreport = sys.exc_info()[2]
fullermessage = traceback.format_tb(exceptionreport)[0]
arcpy.AddError("at this location: \n\n" + fullermessage + "\n")

else:
    # Report error message if Spatial Analyst license is unavailable
    arcpy.AddMessage ("Spatial Analyst license is unavailable")

```



### SHAPEFILE USED



### RESULT SHAPEFILE

## Script 2 : Buffer

“””

THIS SCRIPT APPLIES THE ARCGIS POLYGON TO LINE TOOL

To create an ArcGIS Pro script tool for this script, do the following.

- 1 In Catalog > Toolboxes, select an existing toolbox or create a new one.
- 2 Right-click on the entry for this toolbox in ArcToolbox, and use New > Script to open a dialog box.
- 3 In this dialog box, use Label to name the tool being created and Script File to specify its .py file.
- 4 In the same dialog box, click on Parameters and specify the following before pressing the OK button.

LABEL	DATA TYPE	DIRECTION
Input Shapefile	Feature Layer	Input

- 5 To later revise any of this, right-click to the tool's name and select Properties.

“””

```
# Import external modules
import sys, os, string, arcpy, traceback
from arcpy import env
from arcpy.sa import *

# Allow output file to overwrite any existing file of the same name
arcpy.env.overwriteOutput = True

# Check to see if Spatial Analyst license is available
if arcpy.CheckExtension("spatial") == "Available":

    try:
        # Activate ArcGIS Spatial Analyst license
        arcpy.CheckOutExtension("spatial")

        # Read user inputs from dialog box
        input_shp      = arcpy.GetParameterAsText(0)

        # Output File name
        nameOfOutputShapefile = input_shp + "_Buffer"

        # Buffer
        arcpy.Buffer_analysis(input_shp, "buffer", "200 Kilometers")
        arcpy.CopyFeatures_management("buffer",nameOfOutputShapefile)
        # Add a layer for that new shapefile to the active data frame (which is necessary because
        # there was no user-specified output shapefile with its DIRECTION set to OUTPUT)
        nameOfOutputLayer  = (arcpy.Describe(nameOfOutputShapefile).file)
        arcpy.MakeFeatureLayer_management(nameOfOutputShapefile,nameOfOutputLayer)

        nameOfOutputLayerFile = nameOfOutputShapefile + ".lyrx"
        arcpy.SaveToLayerFile_management(nameOfOutputLayer, nameOfOutputLayerFile, "ABSOLUTE")
```

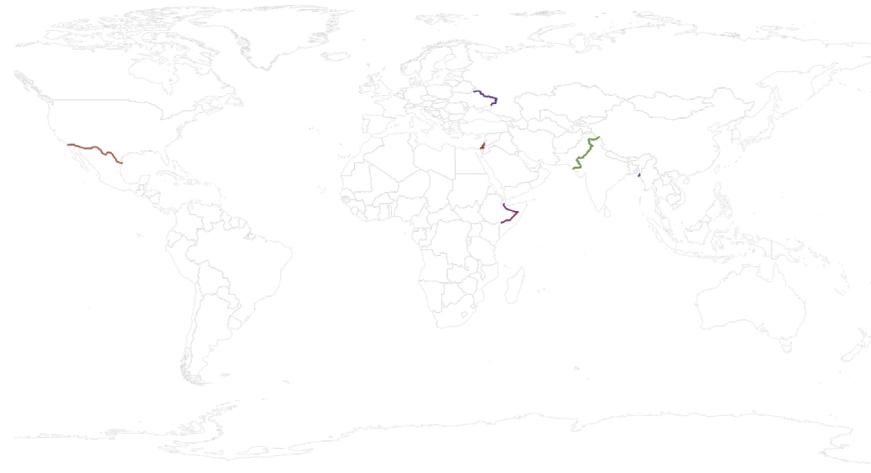
```

currentProject = arcpy.mp.ArcGISProject("CURRENT")
currentMap = currentProject.activeMap
layerFile = arcpy.mp.LayerFile(nameOfFile)
currentMap.addLayer(layerFile,"TOP")

except Exception as e:
    # If unsuccessful, end gracefully by indicating why
    arcpy.AddError('\n' + "Script failed because: \t\t" + e.args[0] )
    # ... and where
    exceptionreport = sys.exc_info()[2]
    fullermessage = traceback.format_tb(exceptionreport)[0]
    arcpy.AddError("at this location: \n\n" + fullermessage + "\n")

else:
    # Report error message if Spatial Analyst license is unavailable
    arcpy.AddMessage ("Spatial Analyst license is unavailable")

```



SHAPEFILE USED



RESULT SHAPEFILE

## Step 2

**Use ArcPro API to reclassify and do all the map algebra on the anthropogenic biomes data. Use GEE to analyze and work on the other data.**

For this process I LANDSAT, MODIS, Night Lighting and SEDAC data.

The steps involved in this process include :

1. Reclassify data
2. Subtract/find change
3. Clip to buffer zones

The SEDAC data is available from 1700 to 2006. The process outlined below is repeated for each of those rasters.

The process is the same in Earth Engine as well. Only to get the borders I am interested in Earth Engine I imported the shapefiles from the previous process and imported that as the data to be used. For the ease to understand the different scripts and programs, the code for GEE will be attached after all the scripts from ArcGIS pro.

As GEE has accessd to more latest data and higher resolution, after the 2000 the data is analyzed for 2000-2010 and 2010-2020. The change isnt as evident but the trend can be seen.

# Script 3 : Reclassify

“””

THIS SCRIPT APPLIES THE ARCGIS SPATIAL ANALYST'S RECLASSIFY TOOL

To create an ArcGIS Pro script tool for this script, do the following.

- 1 In Catalog > Toolboxes, select an existing toolbox or create a new one.
- 2 Right-click on the entry for this toolbox in ArcToolbox, and use New > Script to open a dialog box.
- 3 In this dialog box, use Label to name the tool being created and Script File to specify its .py file.
- 4 In the same dialog box, click on Parameters and specify the following before pressing the OK button.

LABEL	DATA TYPE	DIRECTION
Input Grid	Raster Dataset	Input
Output Grid	Raster Dataset	Output

- 7 To later revise any of this, right-click to the tool's name and select Properties.

In the dialog box for this new tool,"Reclass Instructions" are used to Reclassify existing grid values into new values. These are to be specified as

OldValue newValue; OldValue newValue; OldValue newValue ...

(where each OldValue is an existing value to be replaced by the following newValue)

or as InitialOldValue FinalOldValue newValue

(where newValue replaces all existing values from InitialOldValue through FinalOldValue)  
and where pixels for which no new value is assigned are set to 0.

“””

```
# Import external modules
import sys, os, string, arcpy, traceback
from arcpy import env
from arcpy.sa import *

# Check to see if Spatial Analyst license is available
if arcpy.CheckExtension("spatial") == "Available":

    try:
        # Activate ArcGIS Spatial Analyst license
        arcpy.CheckOutExtension("spatial")

        # Read user inputs from dialog box
        inputGridName    = arcpy.GetParameterAsText(0)
        outputGridName   = arcpy.GetParameterAsText(1)

        # Reclassify
        remap = RemapRange([[10,19,1],[20,29,2],[30,70,3]])

        newGrid         = arcpy.sa.Reclassify(inputGridName,"Value",remap, "NODATA")

        # Save newly created grid
        newGrid.save(outputGridName)
```

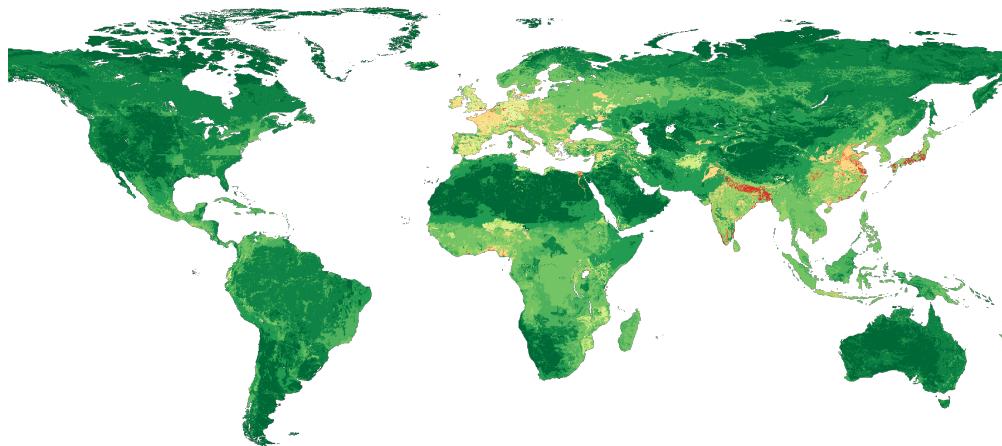
```

# Deactivate ArcGIS Spatial Analyst license
arcpy.CheckInExtension("spatial")

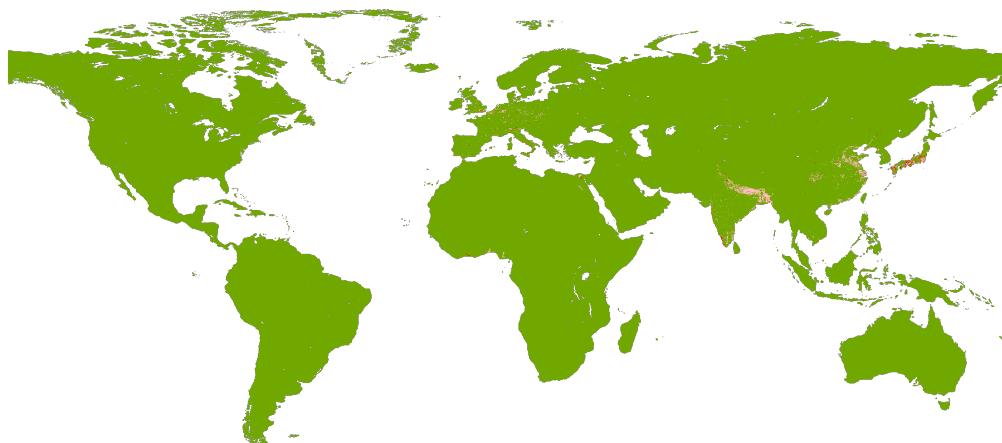
except Exception as e:
    # If unsuccessful, end gracefully by indicating why
    arcpy.AddError("\n" + "Script failed because: \t\t" + e.args[0] )
    # ... and where
    exceptionreport = sys.exc_info()[2]
    fullermessage = traceback.format_tb(exceptionreport)[0]
    arcpy.AddError("at this location: \n\n" + fullermessage + "\n")

else:
    # Report error message if Spatial Analyst license is unavailable
    arcpy.AddMessage ("Spatial Analyst license is unavailable")

```



INPUT RASTER



RECLASSIFIED RASTER

# Script 4 : Raster Calculator

~~~~~

THIS SCRIPT APPLIES THE ARCGIS SPATIAL ANALYST'S MINUS TOOL

To create an ArcGIS Pro script tool for this script, do the following.

- 1 In Catalog > Toolboxes, select an existing toolbox or create a new one.
- 2 Right-click on the entry for this toolbox in ArcToolbox, and use New > Script to open a dialog box.
- 3 In this dialog box, use Label to name the tool being created and Script File to specify its .py file.
- 4 In the same dialog box, click on Parameters and specify the following before pressing the OK button.

| LABEL       | DATA TYPE      | DIRECTION |
|-------------|----------------|-----------|
| Input Grid  | Raster Dataset | Input     |
| Input Grid  | Raster Dataset | Input     |
| Output Grid | Raster Dataset | Output    |

- 7 To later revise any of this, right-click to the tool's name and select Properties.

~~~~~

```
# Import external modules
import sys, os, string, arcpy, traceback
from arcpy import env
from arcpy.sa import *

# Check to see if Spatial Analyst license is available
if arcpy.CheckExtension("spatial") == "Available":

    try:
        # Activate ArcGIS Spatial Analyst license
        arcpy.CheckOutExtension("spatial")

        # Read user inputs from dialog box
        raster1      = arcpy.GetParameterAsText(0)
        raster2      = arcpy.GetParameterAsText(1)
        output_raster = arcpy.GetParameterAsText(2)

        # Raster Calculator
        #newGrid = arcpy.sa.RasterCalculator(["raster1","raster2"], ["r1","r2"], "r1 - r2")

        #newGrid = Hillshade(raster1)
        #newGrid.save(output_raster)

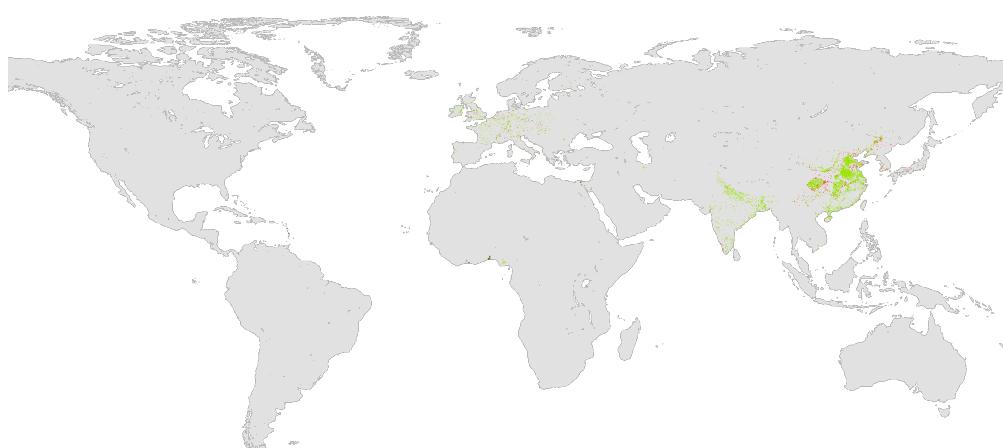
        # Minus operator
        newGrid = Raster(raster1) + Raster(Raster(raster2) * 10)
        newGrid.save(output_raster)

        # Deactivate ArcGIS Spatial Analyst license
        arcpy.CheckInExtension("spatial")
```

```
except Exception as e:  
    # If unsuccessful, end gracefully by indicating why  
    arcpy.AddError("\n" + "Script failed because: \t\t" + e.args[0] )  
    # ... and where  
    exceptionreport = sys.exc_info()[2]  
    fullermessage  = traceback.format_tb(exceptionreport)[0]  
    arcpy.AddError("at this location: \n\n" + fullermessage + "\n")  
  
else:  
    # Report error message if Spatial Analyst license is unavailable  
    arcpy.AddMessage ("Spatial Analyst license is unavailable")
```



INPUT RASTER



SUBTRACTED RASTER

# Script 5 : Clip Function

“””

THIS SCRIPT APPLIES THE ARCGIS POLYGON TO LINE TOOL

To create an ArcGIS Pro script tool for this script, do the following.

- 1 In Catalog > Toolboxes, select an existing toolbox or create a new one.
- 2 Right-click on the entry for this toolbox in ArcToolbox, and use New > Script to open a dialog box.
- 3 In this dialog box, use Label to name the tool being created and Script File to specify its .py file.
- 4 In the same dialog box, click on Parameters and specify the following before pressing the OK button.

LABEL	DATA TYPE	DIRECTION
Input Raster	Raster Dataset	Input
Output Raster	Raster Dataset	Output
Input clipping geometry	Shapefile	Input

- 5 To later revise any of this, right-click to the tool's name and select Properties.

“””

```
# Import external modules
import sys, os, string, arcpy, traceback
from arcpy import env
from arcpy.sa import *

# Allow output file to overwrite any existing file of the same name
arcpy.env.overwriteOutput = True

# Check to see if Spatial Analyst license is available
if arcpy.CheckExtension("spatial") == "Available":

    try:
        # Activate ArcGIS Spatial Analyst license
        arcpy.CheckOutExtension("spatial")

        # Read user inputs from dialog box
        input_raster      = arcpy.GetParameterAsText(0)

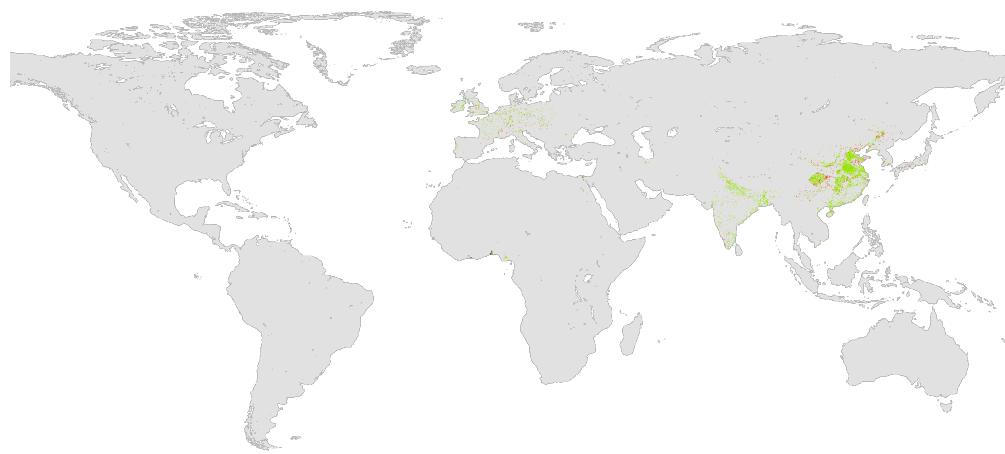
        # Output File name
        nameOfOutputRaster = arcpy.GetParameterAsText(1)

        # Feature Class
        feature_input = arcpy.GetParameterAsText(2)

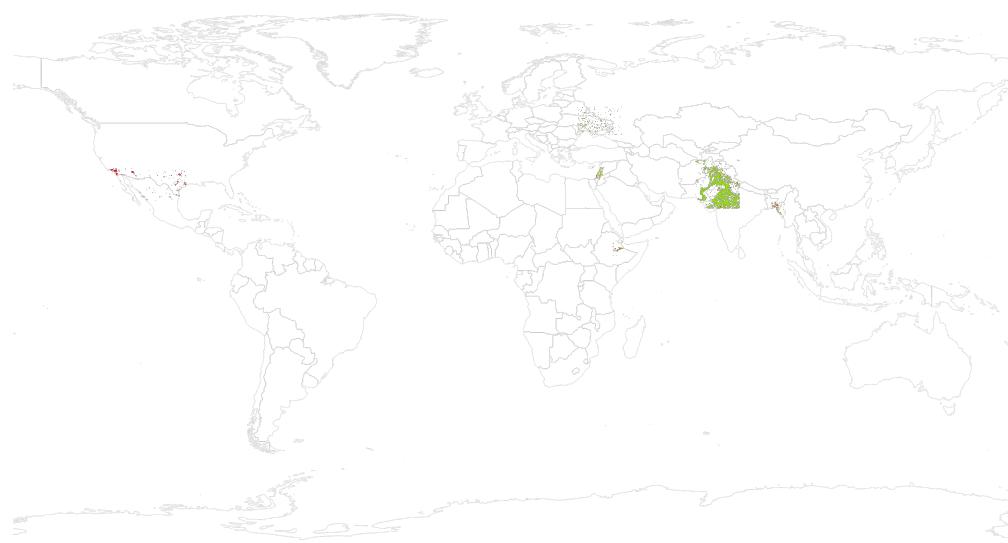
        # Clip
        arcpy.Clip_management(input_raster, "", nameOfOutputRaster, feature_input)

    except Exception as e:
```

```
# If unsuccessful, end gracefully by indicating why  
arcpy.AddError('\n' + "Script failed because: \t\t" + e.args[0] )  
# ... and where  
exceptionreport = sys.exc_info()[2]  
fullermessage  = traceback.format_tb(exceptionreport)[0]  
arcpy.AddError("at this location: \n\n" + fullermessage + "\n")  
  
else:  
    # Report error message if Spatial Analyst license is unavailable  
    arcpy.AddMessage ("Spatial Analyst license is unavailable")
```



INPUT RASTER



CLIPED RASTER

## Script 5 : GEE Code

```
// Final Project
// Palak

/*
var ColorsTemp =[‘aec3d4’, // 00 = evergreen Needleleaf
‘33510d’, // 01 = Evergreen Broadleaf
‘e2720f’, // 02 = Deciduous Needleleaf
‘369b47’, // 03 = Deciduous Broadleaf
‘30eb5b’, // 04 = Mixed Forests
‘387242’, // 05 = Closed Shrublands
‘ff6100’, // 06 = Open Shrubland
‘c3aa69’, // 07 = Woody Savannas
‘b76031’, // 08 = Savannas
‘d9903d’, // 09 = Grassland
‘a9c952’, // 10 = Permanent Wetland
‘111149’, // 11 = Cropland
‘cdb33b’, // 12 = Urban and Built-up Lands:
‘c40f5a’, // 13 = Cropland/Natural Vegetation Mosaics
‘33280d’, // 14 = Permanent Snow and Ice:
‘d7cdcc’, // 15 = Barren
‘2784A7’, // 16 = Water Bodies
].join(“,”);

//Filter image of 2000 from image collection;
var dataset1 =ee.ImageCollection(‘MODIS/006/MCD12Q1’).filter(ee.Filter.date(‘2000-01-01’, ‘2002-01-01’));
var dataset1 = dataset1.first();
//print(“dataset1”, dataset1);

var igbpLandCover2000 = dataset1.select(‘LC_Type1’);
Map.addLayer(igbpLandCover2000, { min:0, max:17, opaque:0.5, palette:ColorsTemp }, ‘IGBP Land Cover2000’);

//Filter image of 2010 from image collection;
var dataset2 =ee.ImageCollection(‘MODIS/006/MCD12Q1’).filter(ee.Filter.date(‘2010-01-01’, ‘2011-01-01’));
var dataset2 = dataset2.first();
//print(“dataset2”, dataset2);

var igbpLandCover2010 = dataset1.select(‘LC_Type1’);
//print(“igbpLandCover2010”, igbpLandCover2010);
Map.addLayer(igbpLandCover2010, { min:0, max:17, opaque:0.5, palette:ColorsTemp }, ‘IGBP Land Cover2010');
```

```

//Filter image of 2019 from image collection;
var dataset3 =ee.ImageCollection('MODIS/006/MCD12Q1').filter(ee.Filter.date('2019-01-01', '2020-01-01'));
var dataset3 = dataset3.first();
//print("dataset3", dataset3);

var igbpLandCover2020 = dataset3.select('LC_Type1');
//print("igbpLandCover2020", igbpLandCover2020);
Map.addLayer(igbpLandCover2020, { min:0, max:17, opaque:0.5, palette:ColorsTemp }, 'IGBP Land Cover2020');

//Remap
var LandCover2000_remap = dataset1.remap([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,1,1,1,1,1,1,1,1,2,3,0,0,0,4], 0,'LC_Type1');
//print("LandCover2000_remap", LandCover2000_remap);
Map.addLayer(LandCover2000_remap,{ min:1, max:4, opaque:0.2, palette:['green','pink','red','blue'] },'Land-Cover2000_remap');

var LandCover2010_remap = dataset2.remap([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,1,1,1,1,1,1,1,1,2,3,0,0,0,4], 0,'LC_Type1');
//print("LandCover2010_remap", LandCover2010_remap);
Map.addLayer(LandCover2010_remap,{ min:1, max:4, opaque:0.7, palette:['green','pink','red','blue'] },'Land-Cover2010_remap');

var LandCover2020_remap = dataset3.remap([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,1,1,1,1,1,1,1,1,2,3,0,0,0,4], 0,'LC_Type1');
//print("LandCover2020_remap", LandCover2020_remap);
Map.addLayer(LandCover2020_remap,{ min:1, max:4, opaque:0.7, palette:['green','pink','red','blue'] },'Land-Cover2020_remap');

// Change
var LandCover_mul = LandCover2010_remap.multiply(10);
var LandCover_change = LandCover_mul.add(LandCover2000_remap);
//print("Landcover Change", LandCover_change);
// Remap to identify areas with land use change, including total land use change;
var Cover_changetotal = LandCover_change.remap([0,11,12,13,14,21,22,23,24,31,32,33,34,41,42,43,44],
[0,0,2,3,4,1,0,3,4,1,2,0,4,1,2,3,0],0);
Map.addLayer (Cover_changetotal, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue'] }, 'Change2010-2000');

var LandCover_mul2 = LandCover2020_remap.multiply(10);
var LandCover_change2 = LandCover_mul2.add(LandCover2010_remap);
//print("Landcover Change", LandCover_change2);
// Remap to identify areas with land use change, including total land use change;

```

```

var Cover_changetotal2 = LandCover_change2.re-
map([0,11,12,13,14,21,22,23,24,31,32,33,34,41,42,43,44],
     [0,0,2,3,4,1,0,3,4,1,2,0,4,1,2,3,0],0);
Map.addLayer (Cover_changetotal2, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','bl
ue']}, 'Change2020-2010' );

//ADDING BOUNDARIES
var ban_myn = ee.FeatureCollection("users/agarwalpal22/BangladeshMynammar");
//print(ban);
Map.addLayer(ban_myn);
var empty = ee.Image().byte();
var ban_Fill = empty.paint({
featureCollection: ban_myn,
})
var banmyn_mask = ban_Fill.eq(0)
var banmyn_buf = banmyn_mask.focal_mode(200000, 'circle', 'meters')
//Map.addLayer(ban_buf)

var ethsom = ee.FeatureCollection("users/agarwalpal22/EthopiaSomalia");
//print(ethsom);
Map.addLayer(ethsom);
var empty = ee.Image().byte();
var ethsom_Fill = empty.paint({
featureCollection: ethsom,
})
var ethsom_mask = ethsom_Fill.eq(0)
var ethsom_buf = ethsom_mask.focal_mode(200000, 'circle', 'meters')
//Map.addLayer(ethsom_buf)

var indpak = ee.FeatureCollection("users/agarwalpal22/IndiaPakistan");
//print(indpak);
Map.addLayer(indpak);
var empty = ee.Image().byte();
var indpak_Fill = empty.paint({
featureCollection: indpak,
})
var indpak_mask = indpak_Fill.eq(0)
var indpak_buf = indpak_mask.focal_mode(200000, 'circle', 'meters')
//Map.addLayer(indpak_buf)

var islpal = ee.FeatureCollection("users/agarwalpal22/IsrealPalestine");
//print(islpal);
Map.addLayer(islpal);
var empty = ee.Image().byte();
var islpal_Fill = empty.paint({
featureCollection: islpal,
})

```

```

var islpal_mask = islpal_Fill.eq(0)
var islpal_buf = islpal_mask.focal_mode(200000, 'circle', 'meters')
//Map.addLayer(islpal_buf)

var islsyr = ee.FeatureCollection("users/agarwalpal22/IsrealSyria");
//print(islsyr);
Map.addLayer(islsyr);
var empty = ee.Image().byte();
var islsyr_Fill = empty.paint({
featureCollection: islsyr,
})
var islsyr_mask = islsyr_Fill.eq(0)
var islsyr_buf = islsyr_mask.focal_mode(1000000, 'circle', 'meters')
//Map.addLayer(islsyr_buf)

var rusukr = ee.FeatureCollection("users/agarwalpal22/RussiaUkraine");
//print(rusukr);
Map.addLayer(rusukr);
var empty = ee.Image().byte();
var rusukr_Fill = empty.paint({
featureCollection: rusukr,
})
var rusukr_mask = rusukr_Fill.eq(0)
var rusukr_buf = rusukr_mask.focal_mode(1000000, 'circle', 'meters')
//Map.addLayer(rusukr_buf)

var usmex = ee.FeatureCollection("users/agarwalpal22/USAMexico");
//print(usmex);
Map.addLayer(usmex);
var empty = ee.Image().byte();
var usmex_Fill = empty.paint({
featureCollection: usmex,
})
var usmex_mask = usmex_Fill.eq(0)
var usmex_buf = usmex_mask.focal_mode(100000, 'circle', 'meters')
Map.addLayer(usmex_buf)

//CHANGE 2010-2000
var change_2010_2000_us = Cover_changetotal.mask(usmex_buf)
Map.addLayer(change_2010_2000_us, { min:0, max:4, opaque:0.7, palette: ['FFFFFF', 'red', 'green', 'pink', 'blue'] }, 'Change2010-2000USMEX' );
var change_2010_2000_ban = Cover_changetotal.mask(banmyn_buf)
Map.addLayer(change_2010_2000_ban, { min:0, max:4, opaque:0.7, palette: ['FFFFFF', 'red', 'green', 'pink', 'blue'] }, 'Change2010-2000BANMY' );
var change_2010_2000_eth = Cover_changetotal.mask(ethsom_buf)
Map.addLayer(change_2010_2000_eth, { min:0, max:4, opaque:0.7, palette: ['FFFFFF', 'red', 'green', 'pink', 'blue'] }, 'Change2010-2000ETHSOM' );

```

```

var change_2010_2000_ind = Cover_changetotal.mask(indpak_buf)
Map.addLayer(change_2010_2000_ind, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2010-2000INDPAK' );
var change_2010_2000_pal = Cover_changetotal.mask(islpal_buf)
Map.addLayer(change_2010_2000_pal, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2010-2000ISLPAL' );
var change_2010_2000_syr = Cover_changetotal.mask(islsyr_buf)
Map.addLayer(change_2010_2000_syr, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2010-2000ISLSYR' );
var change_2010_2000_rus = Cover_changetotal.mask(rusukr_buf)
Map.addLayer(change_2010_2000_rus, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2010-2000RUSUKR' );

// CHANGE 2020-2010
var change_2020_2010_us = Cover_changetotal2.mask(usmex_buf)
Map.addLayer(change_2020_2010_us, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010USMEX' );
var change_2020_2010_ban = Cover_changetotal2.mask(banmyn_buf)
Map.addLayer(change_2020_2010_ban, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010BANMY' );
var change_2020_2010_eth = Cover_changetotal2.mask(ethsom_buf)
Map.addLayer(change_2020_2010_eth, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010ETHSOM' );
var change_2020_2010_ind = Cover_changetotal2.mask(indpak_buf)
Map.addLayer(change_2020_2010_ind, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010INDPAK' );
var change_2020_2010_pal = Cover_changetotal2.mask(islpal_buf)
Map.addLayer(change_2020_2010_pal, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010ISLPAL' );
var change_2020_2010_syr = Cover_changetotal2.mask(islsyr_buf)
Map.addLayer(change_2020_2010_syr, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010ISLSYR' );
var change_2020_2010_rus = Cover_changetotal2.mask(rusukr_buf)
Map.addLayer(change_2020_2010_rus, { min:0, max:4, opaque:0.7, palette: ['FFFFFF','red','green','pink','blue']}, 'Change2020-2010RUSUKR' );

*/
// NDVI index (mapping from 1980-2010)

var ndvi_1980 = ee.ImageCollection("NASA/GIMMS/3GV0").filter(ee.Filter.date('1981-07-01', '1990-01-01'));
var ndvi_1980 = ndvi_1980.first();
var ndvi_1980 = ndvi_1980.select('ndvi');
Map.addLayer(ndvi_1980, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'NDVI_1980');

var ndvi_1990 = ee.ImageCollection("NASA/GIMMS/3GV0").filter(ee.Filter.date('1990-01-01', '2000-01-01'));

```

```

var ndvi_1990 = ndvi_1990.first();
var ndvi_1990 = ndvi_1990.select('ndvi');
Map.addLayer(ndvi_1990, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'NDVI_1990');

var ndvi_2000 = ee.ImageCollection("NASA/GIMMS/3GV0").filter(ee.Filter.date('2000-01-01', '2010-01-01'));
var ndvi_2000 = ndvi_2000.first();
var ndvi_2000 = ndvi_2000.select('ndvi');
Map.addLayer(ndvi_2000, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'NDVI_2000');

var ndvi_8090 = ndvi_1990.subtract(ndvi_1980);
print(ndvi_8090);
Map.addLayer(ndvi_8090, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990')

var ndvi_9020 = ndvi_2000.subtract(ndvi_1990);
print(ndvi_9020);
Map.addLayer(ndvi_9020, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1990/2000')

//CHANGE 1980-1990
var change_1980_1990_us = ndvi_8090.mask(usmex_buf)
Map.addLayer(change_1980_1990_us, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990USMEX' );
var change_1980_1990_ban = ndvi_8090.mask(banmyn_buf)
Map.addLayer(change_1980_1990_ban, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990BANMY' );
var change_1980_1990_eth = ndvi_8090.mask(ethsom_buf)
Map.addLayer(change_1980_1990_eth, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990ETHSOM' );
var change_1980_1990_ind = ndvi_8090.mask(indpak_buf)
Map.addLayer(change_1980_1990_ind, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990INDPAK' );
var change_1980_1990_pal = ndvi_8090.mask(islpal_buf)
Map.addLayer(change_1980_1990_pal, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990ISLPAL' );
var change_1980_1990_syr = ndvi_8090.mask(islsyr_buf)
Map.addLayer(change_1980_1990_syr, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990ISLSYR' );
var change_1980_1990_rus = ndvi_8090.mask(rusukr_buf)
Map.addLayer(change_1980_1990_rus, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1980/1990RUSUKR' );

//CHANGE 1990-2000
var change_1990_2000_us = ndvi_9020.mask(usmex_buf)
Map.addLayer(change_1990_2000_us, { min:-1, max:1, opaque:0.7, palette: ['blue','white','green']}, 'Change_NDVI_1990/2000USMEX' );

```

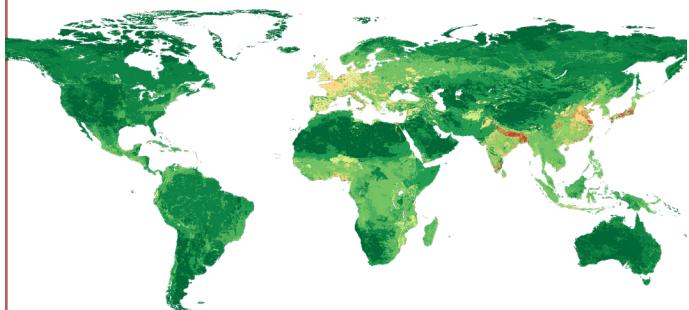
```
var change_1990_2000_ban = ndvi_9020.mask(banmyn_buf)
Map.addLayer(change_1990_2000_ban, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000BANMY')
var change_1990_2000_eth = ndvi_9020.mask(ethsom_buf)
Map.addLayer(change_1990_2000_eth, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000ETHSOM')
var change_1990_2000_ind = ndvi_9020.mask(indpak_buf)
Map.addLayer(change_1990_2000_ind, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000INDPAK')
var change_1990_2000_pal = ndvi_9020.mask(islpal_buf)
Map.addLayer(change_1990_2000_pal, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000ISLPAL')
var change_1990_2000_syr = ndvi_9020.mask(islsyr_buf)
Map.addLayer(change_1990_2000_syr, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000ISLSYR')
var change_1990_2000_rus = ndvi_9020.mask(rusukr_buf)
Map.addLayer(change_1990_2000_rus, { min:-1, max:1, opaque:0.7, palette: ['blue","white","green']},  
'Change_NDVI_1990/2000RUSUKR')
```

# Results

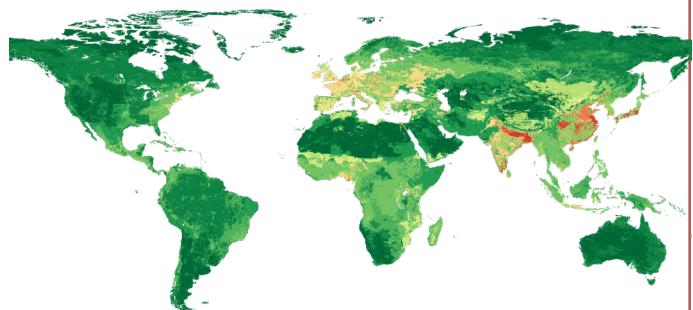
The results are represented in the following order:

1. Urban growth around the world using SEDAC data
2. Increase in urban development using Land use data
3. Decrease in vegetation data

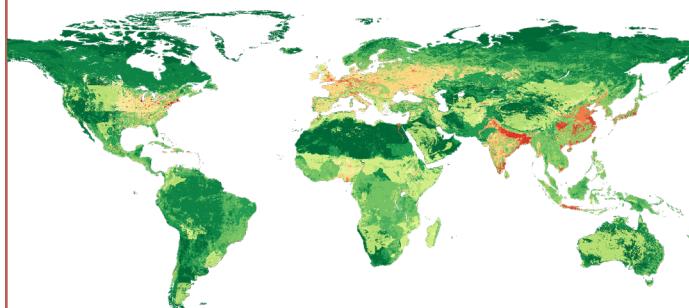
First lets look at these datas on the global scale



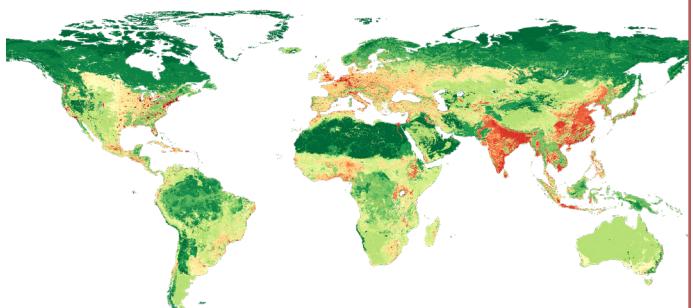
1700



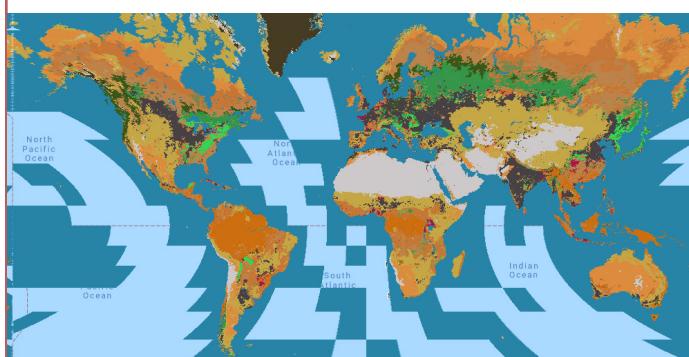
1800



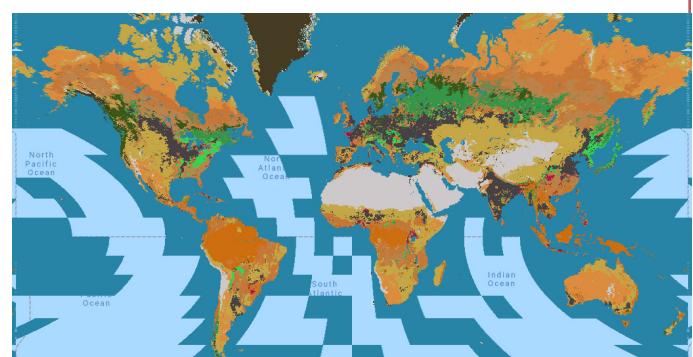
1900



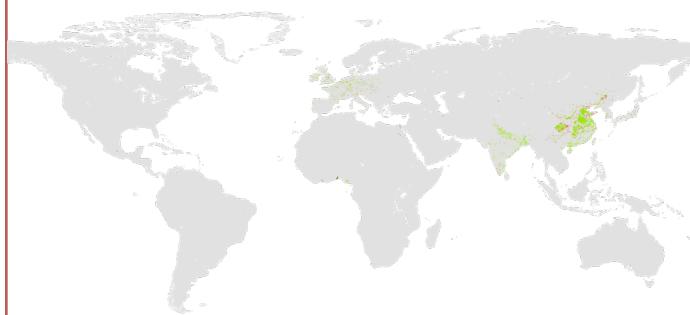
2000



2010



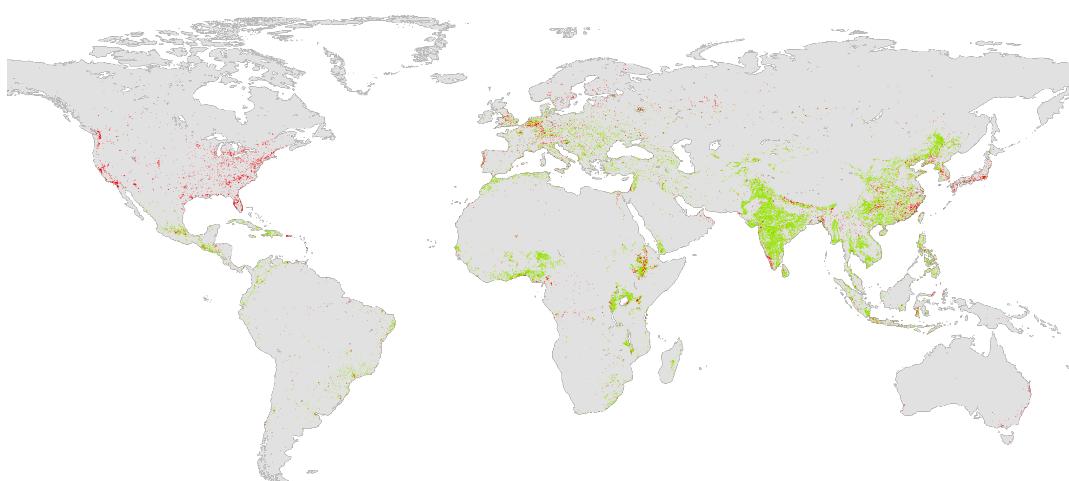
2020



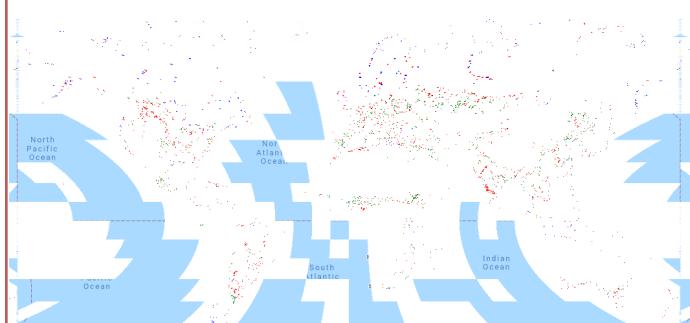
1800-1700



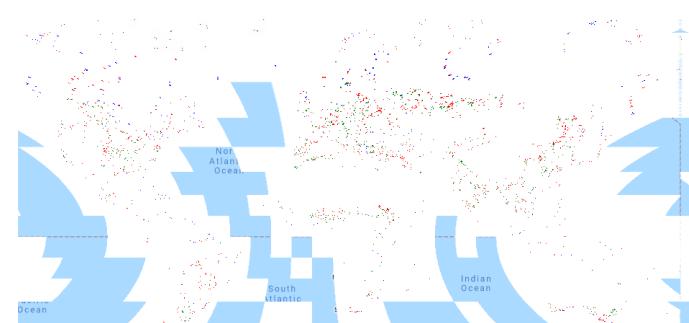
1900-1800



2000-1900



2000-2010

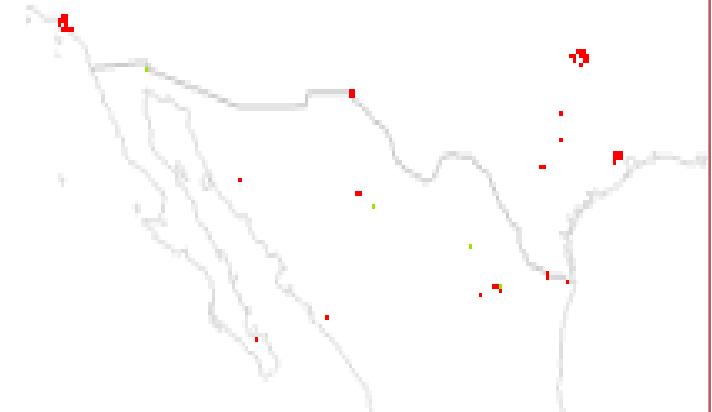


2010-2020

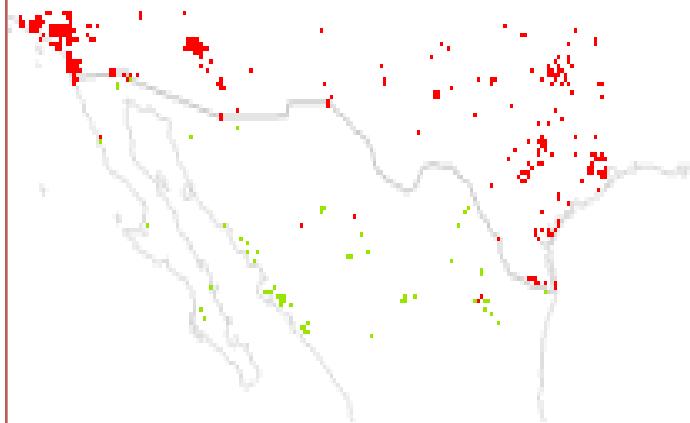
# US-MEXICO BORDER



1800-1700



1900-1800



2000-1900

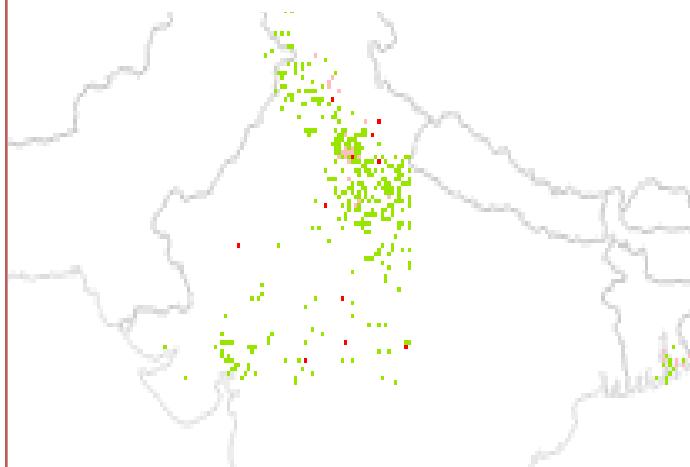


2010-2000

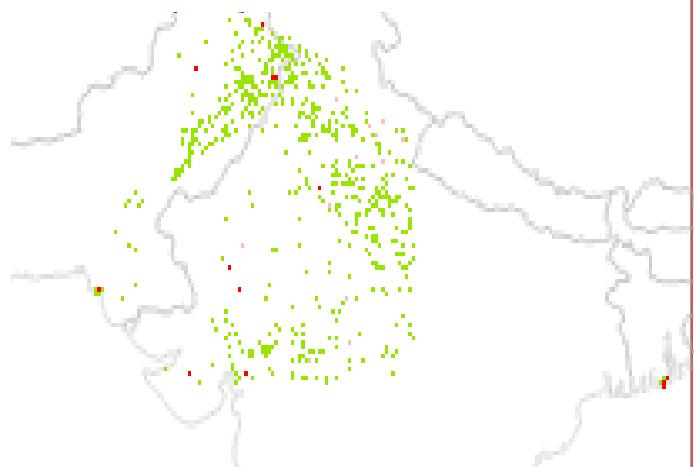


2020-2010

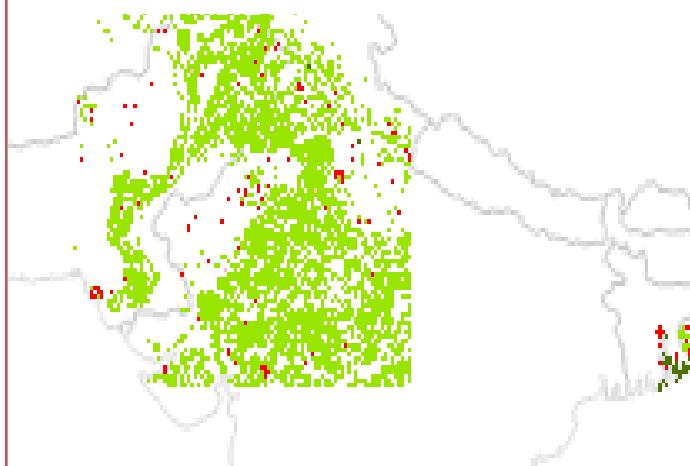
# INDIA-PAKISTAN BORDER



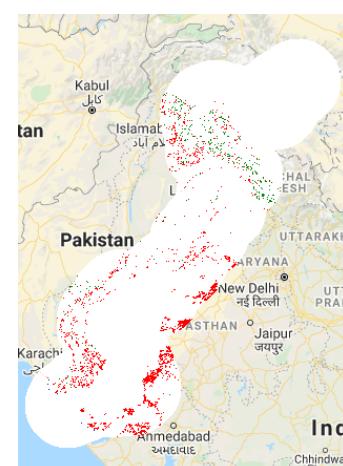
1800-1700



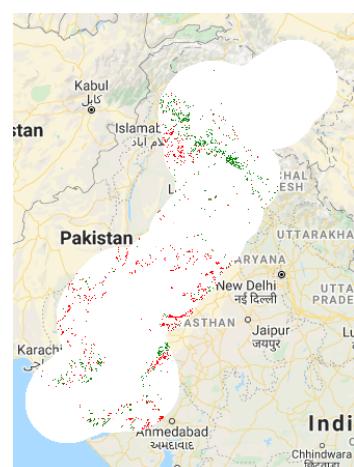
1900-1800



2000-1900

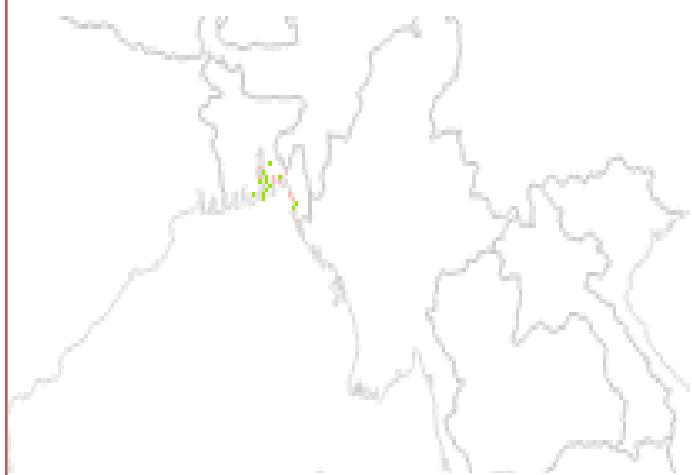


2010-2000

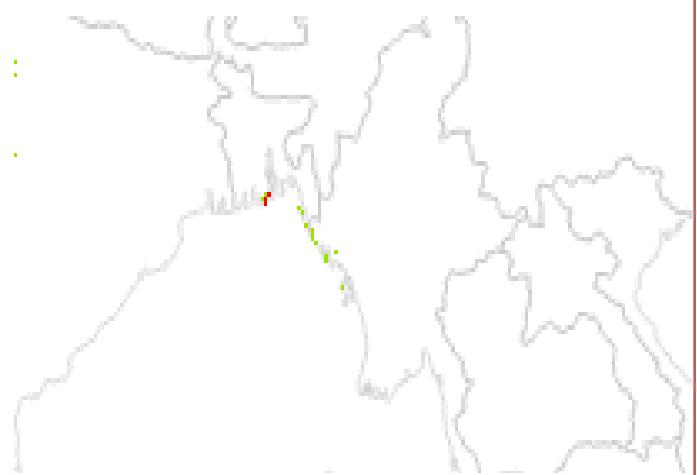


2020-2010

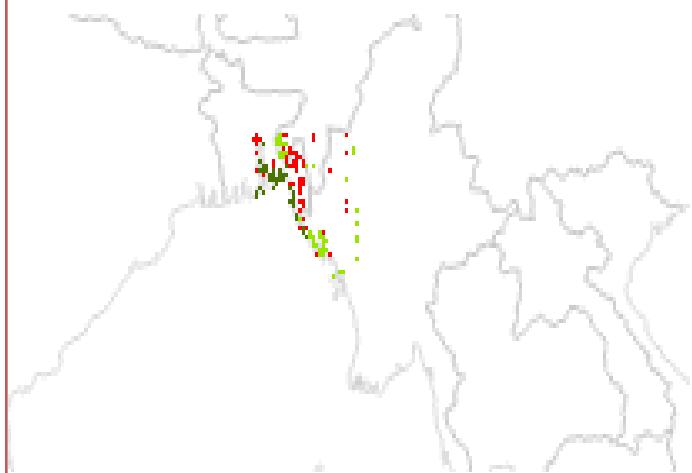
# BANGLADESH-MYANMAR BORDER



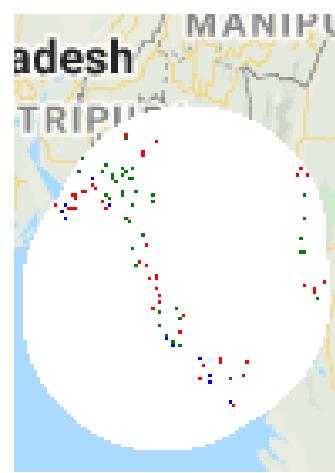
1800-1700



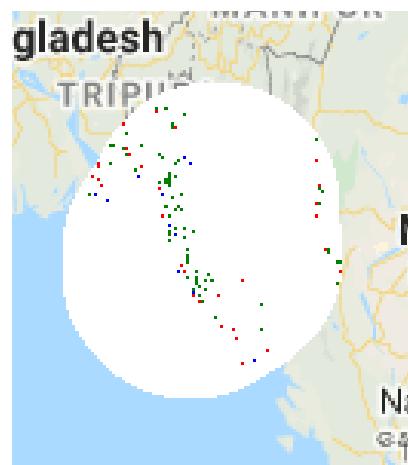
1900-1800



2000-1900

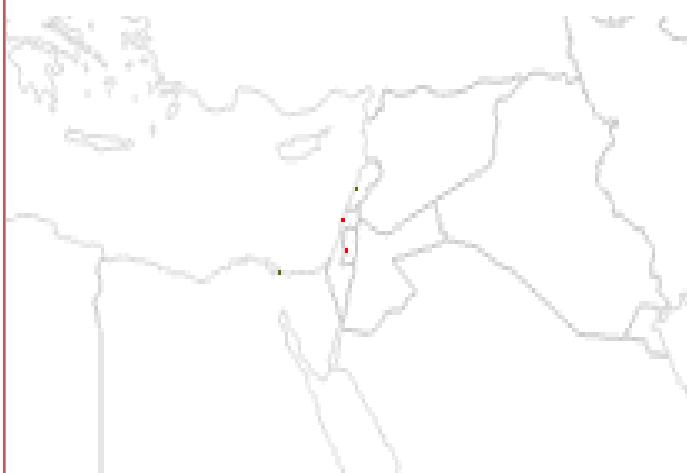


2010-2000

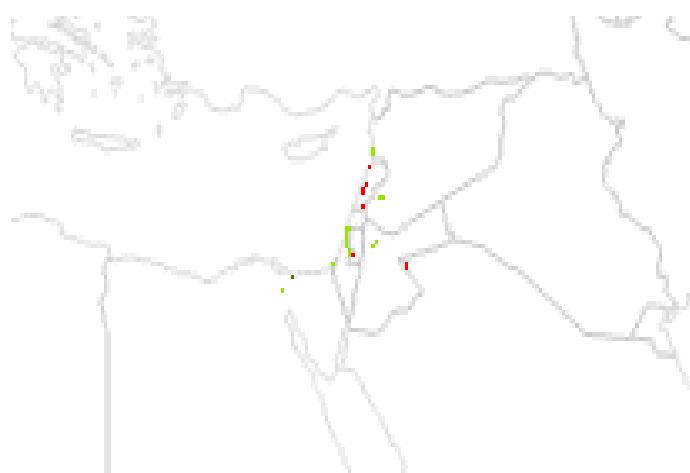


2020-2010

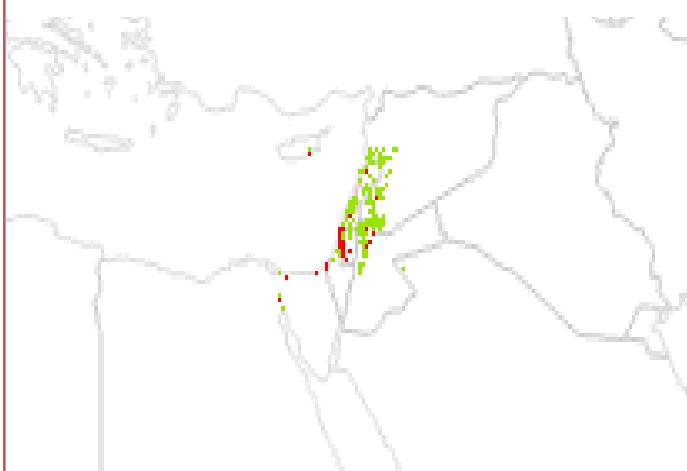
# ISREAL-PALESTINE AND ISREAL-SYRIA BORDER



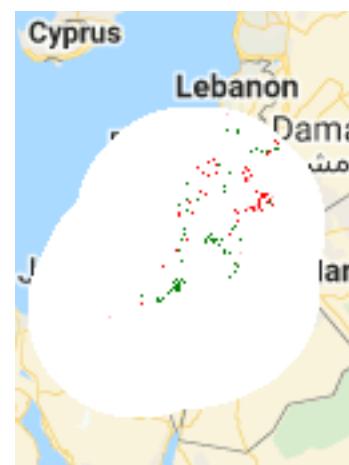
1800-1700



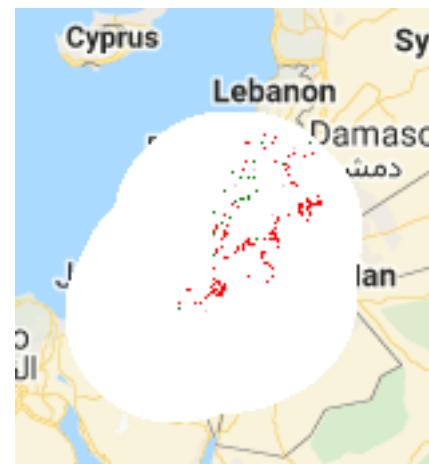
1900-1800



2000-1900



2010-2000



2020-2010

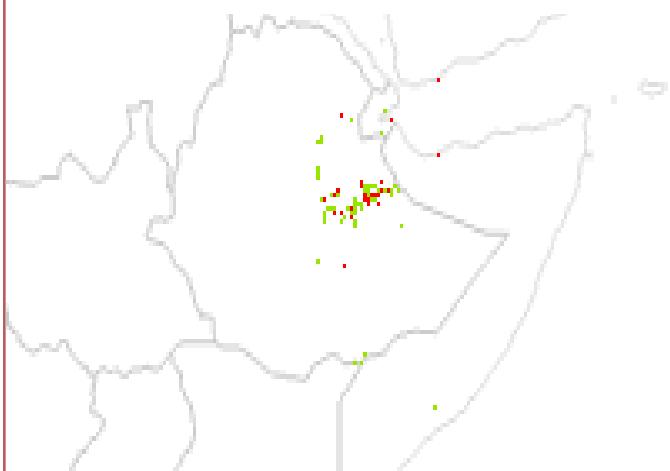
# ETHOPIA-SOMALIA BORDER



1800-1700



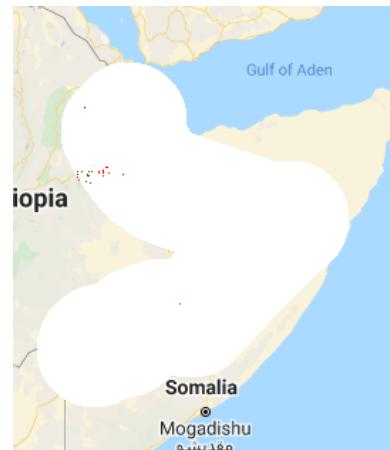
1900-1800



2000-1900



2010-2000

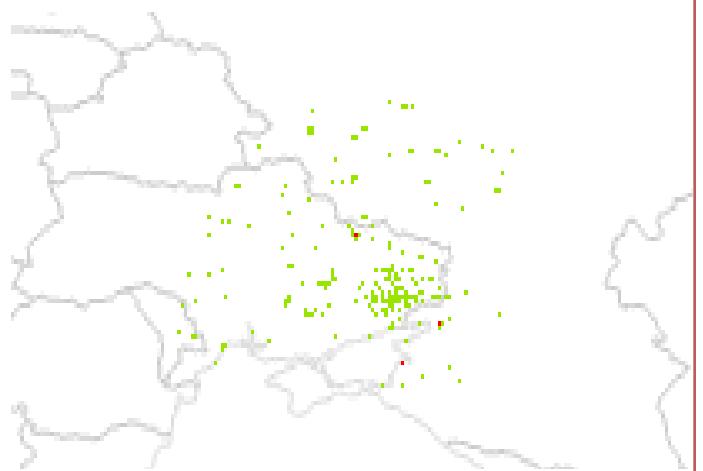


2020-2010

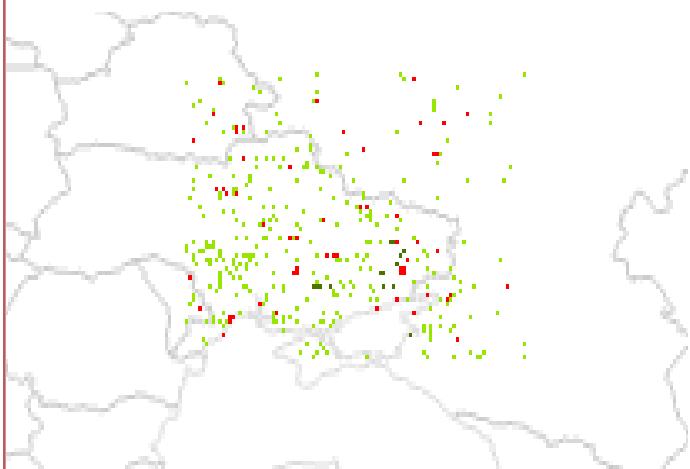
# RUSSIA-UKRAINE BORDER



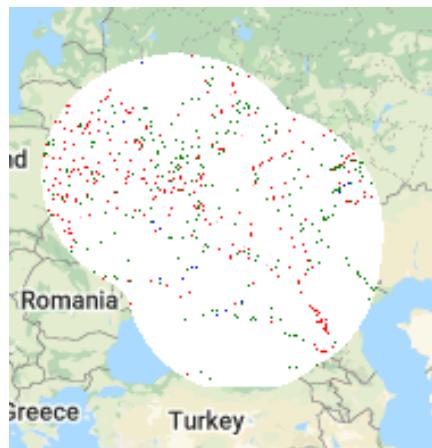
1800-1700



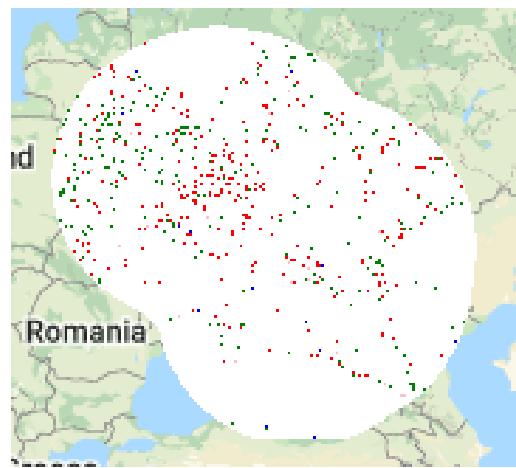
1900-1800



2000-1900

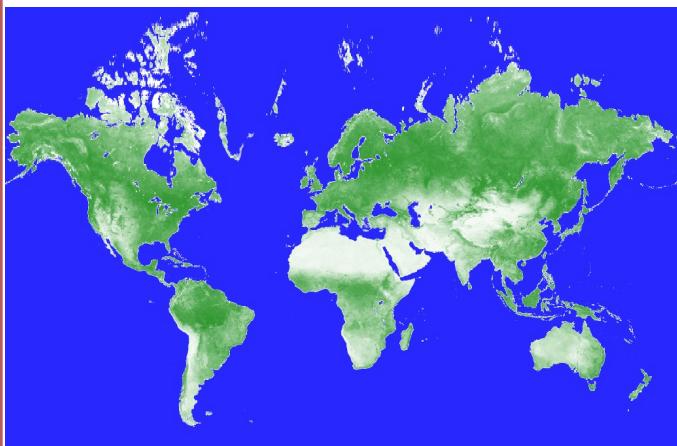


2010-2000

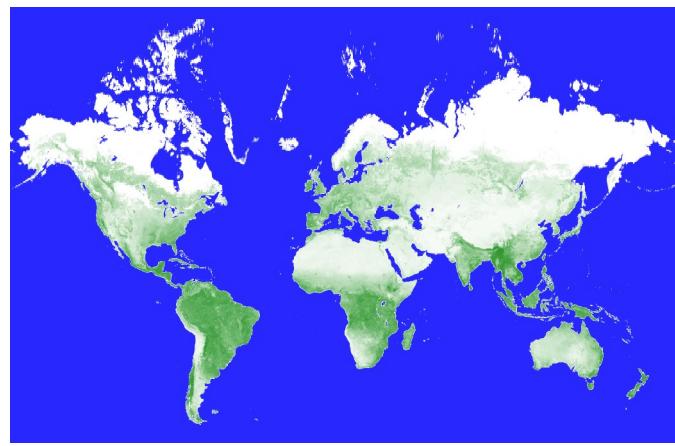


2020-2010

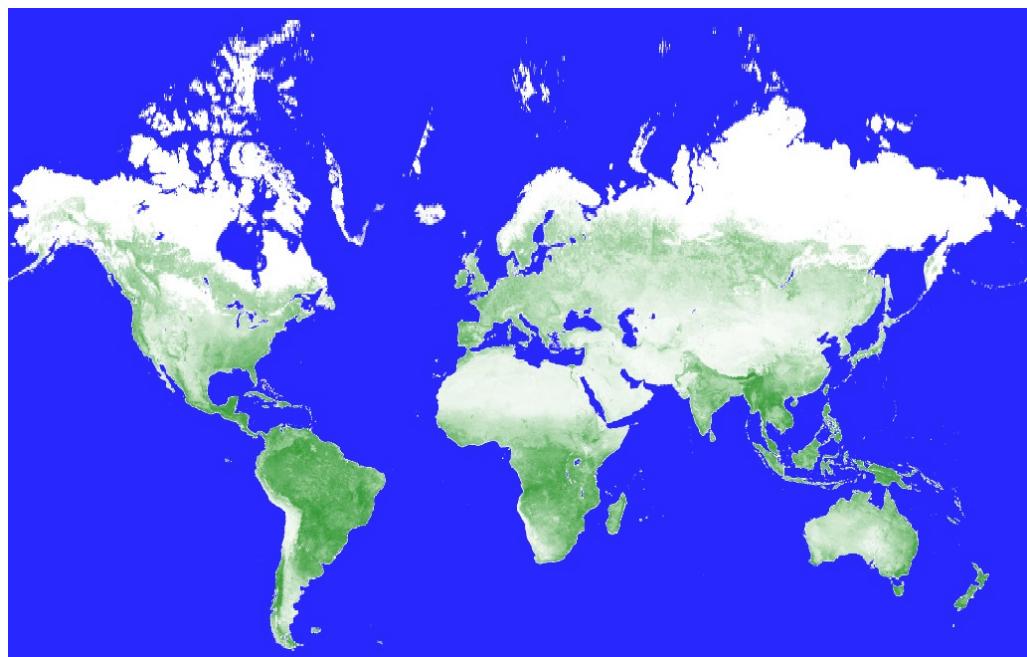
# NDVI Index and Change



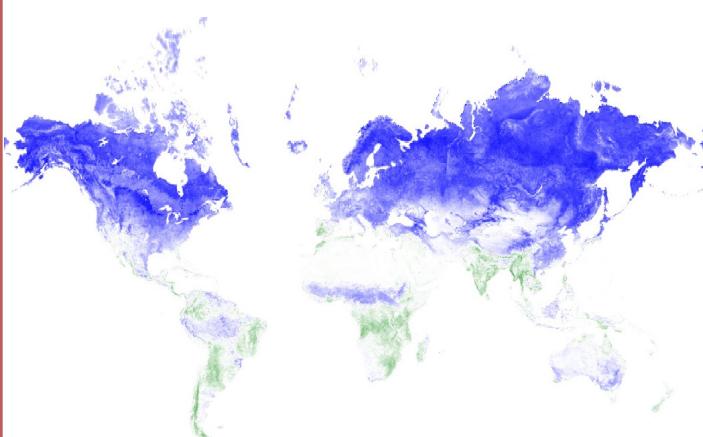
1980



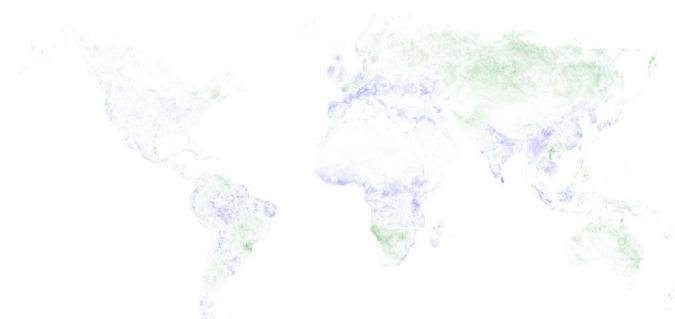
1990



2000



Change  
1980-1990



Change  
1990-2000

Thank you