Shrinking Cities in China

Analysis of Urban Constructed Area & Population Change in Past Decade

Geospatial Software Design Final Project
Professor Dana Tomlin
Google Earth Engine & Arcpy
Shu Xingxing

1.Introduction

In China, the phenomenon of urban shrinkage is still ignored by the mainstream of the society. In the overall planning of cities across the country, few cities have made plans to cope with the decline in population. Some small cities still paint unrealistic blueprints for population growth. However, the decline of the export processing industry triggered by the global financial crisis and the arrival of the Lewis turning point of the domestic labor force have changed the conditions for China's urban growth.

This study is based on night light data and Chinese census data to study the shrinkage of counties or cities in China. Since the census data are only available for 2000 and 2010, other data are selected accordingly.

In the first part, I processed the nighttime light data and determined the light threshold based on the empirical comparison method.

The second part is to compare population change during this decade. I subtracted population map of 2010 by 2000 to see spatial distribution of population loss.

The third part is in Arcpy with some exploratory analysis.

2.Data Resources

Raster Data:

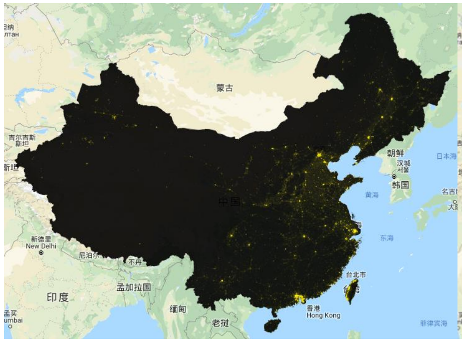
1.Nighttime Light:
https://developers.google.com/earth-engine/datasets/catalog/NOAA_DMSP
OLS_CALIBRATED_LIGHTS_V4

2.Landcover Data:
https://developers.google.com/earth-engine/datasets/catalog/MODIS_051_MCD12Q1

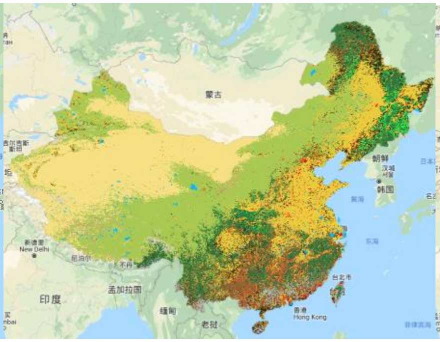
3.Population Data:
Population raster of 2000 & 2010 of China based on Census Data

Vector Data:

1. Boundaries: Nation, Provinces, counties and Jiedao (the basic administrative unit of a city proper in China).
2. Statistical Data: Urban Constructed Area on China's Urban Yearbook



Nighttime Light Map of 2000



Landcover Map of 2000

Population Map of 2000

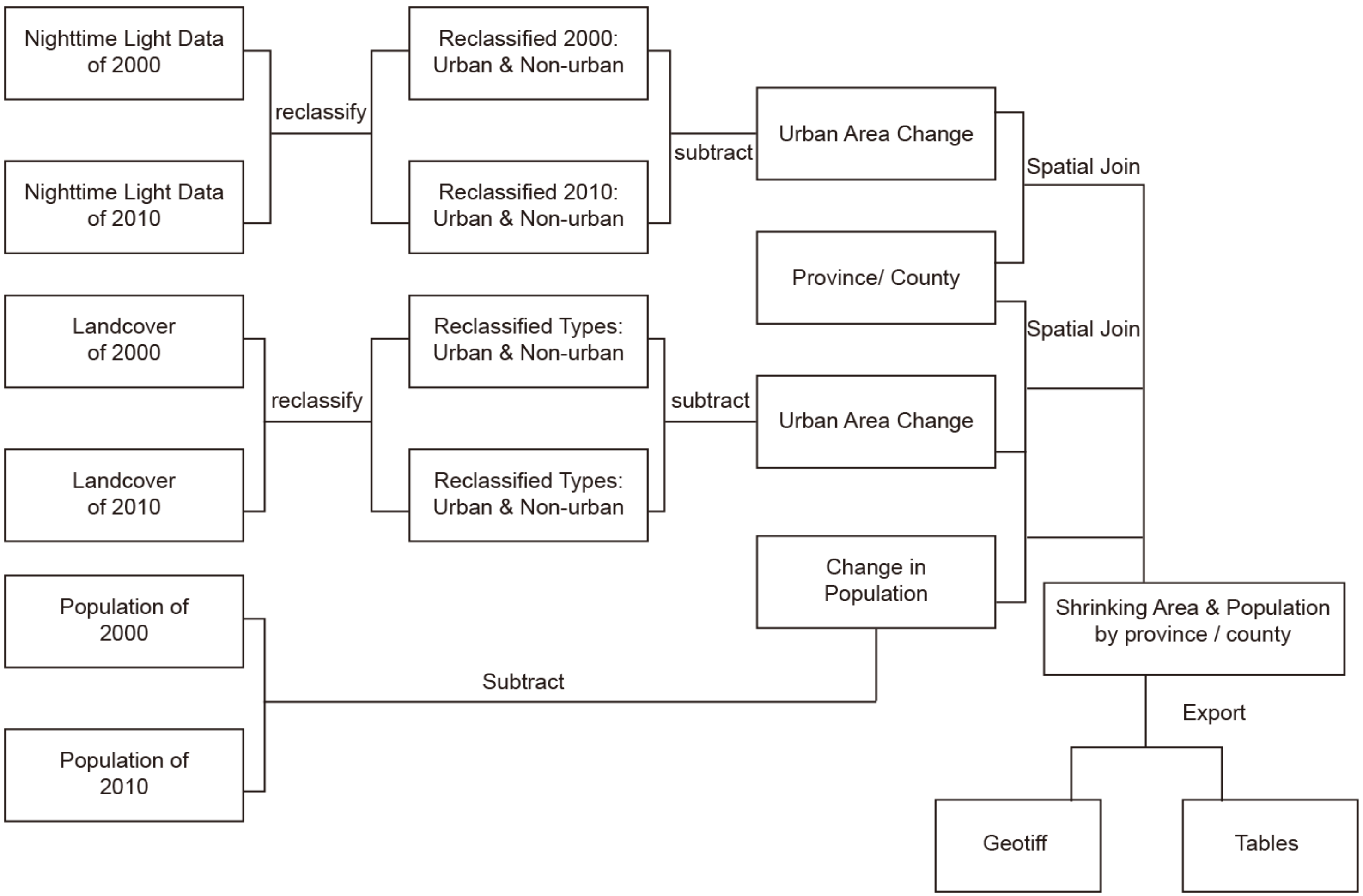
Provinces of China

3. Workflow - GEE

Meet Earth Engine

Google Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities and makes it available for scientists, researchers, and developers to detect changes, map trends, and quantify differences on the Earth's surface.

from https://earthengine.google.com/



3. Workflow - ArcPy

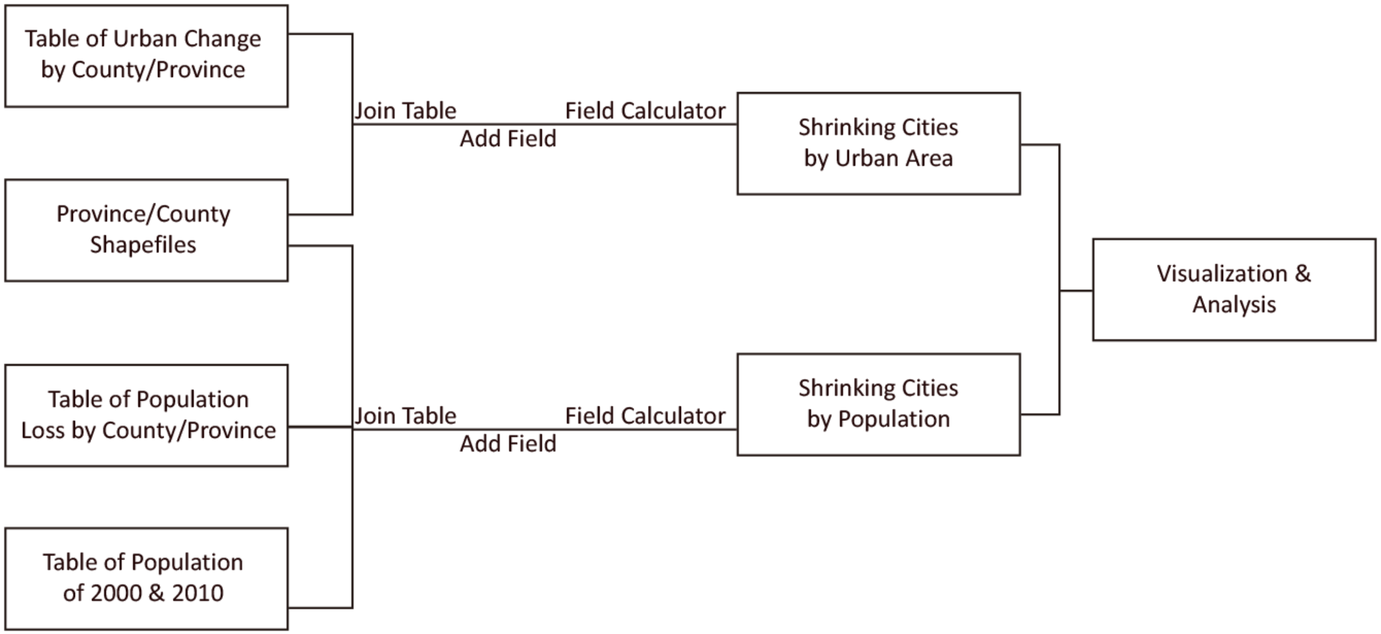
What is ArcPy?

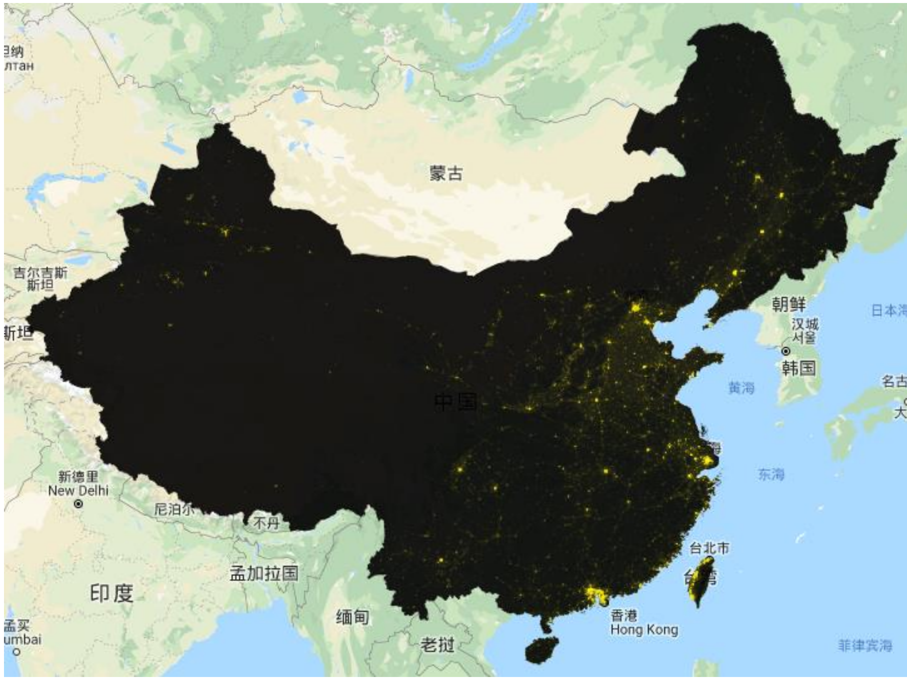
ArcPy is a Python site package that provides a useful and productive way to perform geographic data analysis, data conversion, data management, and map automation with Python.

This package provides a rich and native Python experience offering code completion (type a keyword and a dot to get a pop-up list of properties and methods supported by that keyword; select one to insert it) and reference documentation for each function, module, and class.

The additional power of using ArcPy is that Python is a general-purpose programming language. It is interpreted and dynamically typed and is suited for interactive work and quick prototyping of one-off programs known as scripts while being powerful enough to write large applications in. ArcGIS applications written with ArcPy benefit from the development of additional modules in numerous niches of Python by GIS professionals and programmers from many different disciplines.

from https://pro.arcgis.com/en/pro-app/arcpy/get-started/what-is-arcpy-.htm





Nighttime Light Map of 2000

Nighttime Light Map of 2010

4. Change of Urban Area

4.1 Changes based on Nighttime Light

First, I use nighttime data to determine the urbanized area. To determine the urban region, light threshold need to be set properly. The most commonly used method for determining the threshold is the empirical comparison method. That is, the area of the urban built-up area under different light thresholds is calculated, and then compared with the light data in the city statistical yearbook. The smallest difference is used as the appropriate threshold.

Workflow of Determine Light Threshold

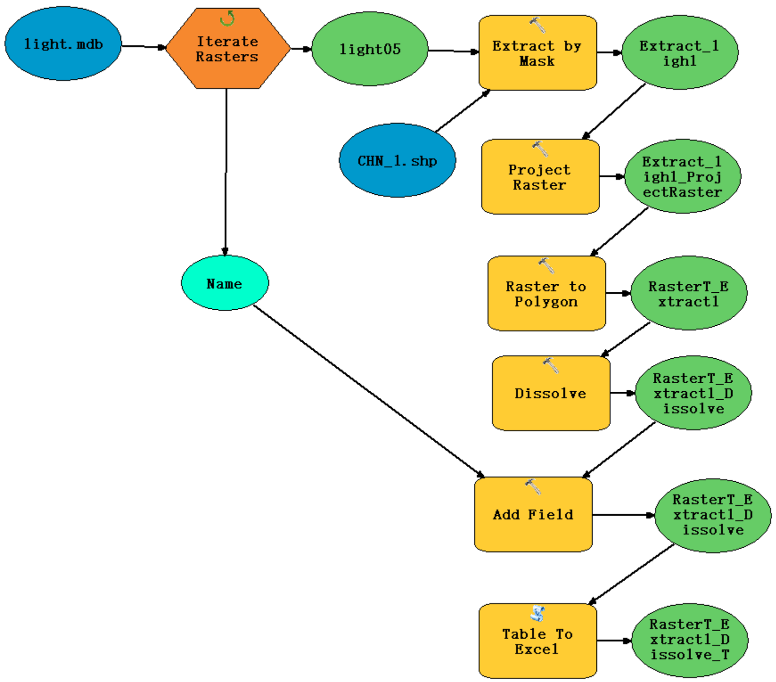


Table of Analysis

Light Threshold	2000 (km2) Urban area under each threshold	2010 (km2) Urban area under each threshold	2000 (km2) Cumulative sum of urban area	2010 (km2) Cumulative sum of urban area	2000 (km2)abs of difference to official data	2010 (km2)abs of difference to official data	2000 (km2) official urban area	2010 (km2) official urban area
63	2841.878159	17070.43158	2841.878159	17070.43158	29678.82184	166345.6684	32520.7	183416.1
62	2807.774814	16499.48813	5649.652973	33569.91971	26871.04703	149846.1803		
61	2647.14878	12667.53666	8296.801753	46237.45637	24223.89825	137178.6436		
60	2731.919549	10386.98259	11028.7213	56624.43896	21491.9787	126791.661		
59	2704.04008	8874.700699	13732.76138	65499.13966	18787.93862	117916.9603		
58	2689.908764	8157.563585	16422.67014	73656.70325	16098.02986	109759.3968		
57	2640.020378	7373.4154	19062.69052	81030.11865	13458.00948	102385.9814		
56	2433.273462	6807.764135	21495.96398	87837.88278	11024.73602	95578.21722		
55	2344.492662	6346.81382	23840.45665	94184.6966	8680.243353	89231.4034		
54	2325.056883	6129.302353	26165.51353	100313.999	6355.18647	83102.10105		
53	2177.681951	6219.587217	28343.19548	106533.5862	4177.504518	76882.51383		
52	2287.925506	5836.545878	30631.12099	112370.132	1889.579012	71045.96795		
51	2179.290711	5687.889039	32810.4117	118058.0211	289.7116989	65358.07891		
50	2203.102783	5660.158773	35013.51448	123718.1799	2492.814482	59697.92014		
49	2127.403509	5511.341724	37140.91799	129229.5216	4620.21799	54186.57842		
48	2159.218367	5380.716207	39300.13636	134610.2378	6779.436358	48805.86221		
47	2178.114623	5467.970369	41478.25098	140078.2082	8957.55098	43337.89184		
46	2188.838231	5336.369924	43667.08921	145414.5781	11146.38921	38001.52192		
45	2207.421705	5371.838432	45874.51092	150786.4165	13353.81092	32629.68348		
44	2205.218978	5538.161982	48079.72989	156324.5785	15559.02989	27091.5215		
43	2385.010392	5514.849377	50464.74029	161839.4279	17944.04029	21576.67212		
42	2383.551028	5645.7149	52848.29131	167485.1428	20327.59131	15930.95722		
41	2429.018234	5644.161722	55277.30955	173129.3045	22756.60955	10286.7955		

Then I did the analysis in Excel. I calculated the accumulated sum of urban area at each threshold and calculated the absolute difference with official data which is 32520.7 Km2 in 2000 & 183416.1 Km2 in 2010. And find out the threshold of 20 in 2000 and 39 in 2010 have the least difference with official data.

Reclassified Urban Area of 2000

Reclassified Urban Area of 2010

```
// To determine the light threshold, I use ArcGIS to calculate the urban area under 63 thresholds and
// compare them to official data from China's Urban Yearbook of 2001 & 2011.
// Based on analysis in ArcGIS, I define the areas with nightlight values larger than 15
// as urban areas in 2000 and areas with nightlight values larger than 30 as urban areas in 2010.
// Then reclassify the urban areas in 2000 and 2010 respectively (Urban area:1 & non-urban area :0)

var urban2000 = light2000.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20],
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], 1, 'stable_lights' ).clip(CHN);
var urban2010 = light2010.remap( [39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63],
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1], 0, 'stable_lights' ).clip(CHN);
Map.addLayer(urban2000,{min:0,max:1,palette:['000000','ff1900'],opacity:0.9},'urban2000')
Map.addLayer(urban2010,{min:0,max:1,palette:['000000','ff1900'],opacity:0.9},'urban2010')

// Subtract reclassified raster of 2000 by reclassified raster of 2010 to get the three types of change
// in urban areas: 1 - developed; 0 - unchanged; -1 - shrank.
var change_urban = urban2010.subtract(urban2000)
Map.addLayer (change_urban, {min:-1,max:1,palette:['85f7ff','000000','ff0066'],opacity:0.9}, 'shrink' );

// Select shrinking rasters
var shrink_urban = change_urban.gt(-1)
```

Change in Urban Area (2010 - 2000)

After reclassified the nighttime map to two categories: urban area with value 1 and non-urban area with value 0.

I subtracted the reclassified map of 2000 by the map of 2010 and get the map of change.

Areas with value 0 means no changes in the past decade; Areas with value 1 means increase in urban area and values of -1 means shrinkage of urban area.

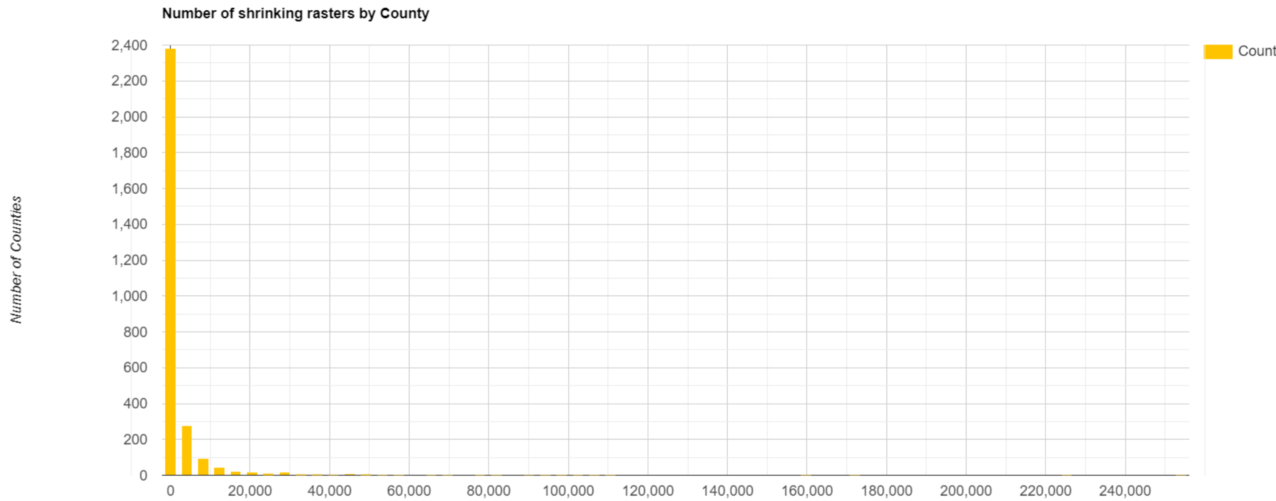
As shows on the left map, red areas means urban expansion and blue areas means urban shrinking.

The shrinking areas are mostly located at northeastern part in Heilongjiang, Jilin and Liaoning Province.

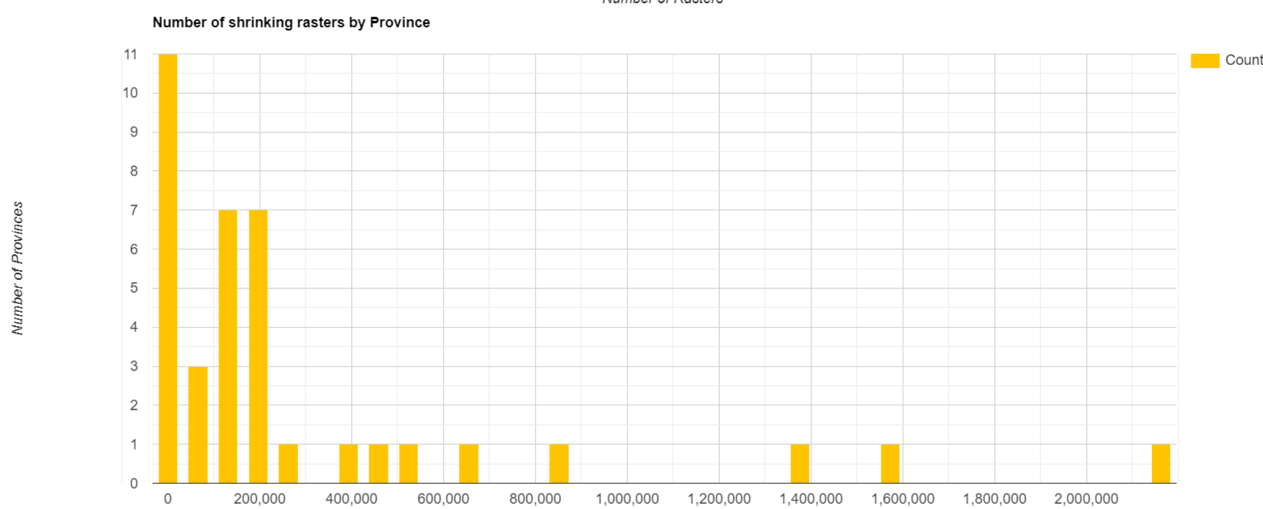
In the Shanxi Province and southwestern part in Yunnan.

Also in far west Xinjiang and middle part of Hubei Province.

```
// Select shrinking rasters
var shrink_urban = change_urban.gt(-1)

// Aggregate shrinking area into provinces
// get the sum of raster number of decreased areas of each province
var province_shrink = shrink_urban.reduceRegions({
  collection: province,
  reducer:ee.Reducer.sum(),
  scale:1000
})
print(ee.Feature(province_shrink.first()),'province_shrink');

// get the sum of raster number of decreased areas of each county
var county_shrink = shrink_urban.reduceRegions({
  collection: county,
  reducer:ee.Reducer.sum(),
  scale:1000
})
print(ee.Feature(county_shrink.first()),'province_shrink');

// Make histogram of the shrinking areas by province
var charts_pr = Chart.feature.histogram(province_shrink, 'sum',50)
var charts_pr = charts_pr.setOptions({ title:'Number of shrinking rasters by Province',
colors : ['ffc400'],
hAxis: {title:'Number of Rasters'},
vAxis: {title:'Number of Provinces'}
});
print(charts_pr)

// Make histogram of the shrinking areas by county
var charts_ct = Chart.feature.histogram(county_shrink, 'sum',50)
var charts_ct = charts_ct.setOptions({ title:'Number of shrinking rasters by County',
colors : ['ffc400'],
hAxis: {title:'Number of Rasters'},
vAxis: {title:'Number of Counties'}
});
print(charts_ct)
```



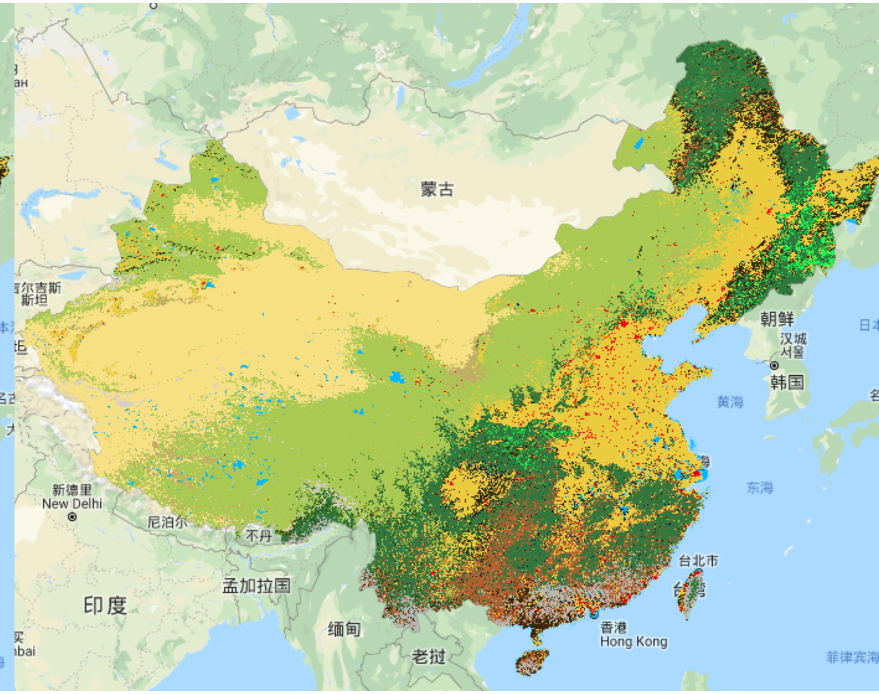
Histogram of Shrinking pixels by County



Histogram of Shrinking pixels by Province

11 out of 36 provinces do not have shrank urban area and 80% have less than 200,000 shrank raster units(1036*1036m). As for county level, most counties do not have shrinking pixels and most have number of shrinking unit (1036*1036m) less than 50.

Landcover of 2000



Landcover of 2010

```
// Part2 --- Use Landcover data to determine the change in urban areas.
// To be conservative, I use landcover data to determine the urban areas just in case.

// Import the land cover image for 2000 and 2010
var landuse2010_original = ee.Image('MODIS/051/MCD12Q1/2010_01_01').select('Land_Cover_Type_1').clip(CHN);
var landuse2000_original = ee.Image('MODIS/051/MCD12Q1/2001_01_01').select('Land_Cover_Type_1').clip(CHN);

// add colors based on class example
var ColorsForMODIS = ['00b7ff', // 00 = Water
'33510d', // 01 = Forest
'b8b6b4', // 02 =
'3b9c4b', // 03 = Deciduous Needleleaf Forest
'20f551', // 04 = Deciduous Broadleaf Forest
'32783e', // 05 = Mixed Deciduous Forest
'ff6100', // 06 = Closed Shrubland
'c3aa69', // 07 = Open Shrubland
'b76031', // 08 = Woody Savanna
'd9903d', // 09 = Savanna
'a9c952', // 10 = Grassland
'111149', // 11 = Permanent Wetland
'ebcc3f', // 12 = Cropland
'f0000c', // 13 = Urban
'33280d', // 14 = Crops & Natural Vegetation
'd7cdcc', // 15 = Permanent Snow & Ice
'f7e084', // 16 = Barren / Desert
'6f6f6f' //17 = Unclassified
].join(',');
```

4.2 Changes based on Landcover Map

To be conservative, I use landcover data of 2000 and 2010 to extract the shrank urban area as a reference.

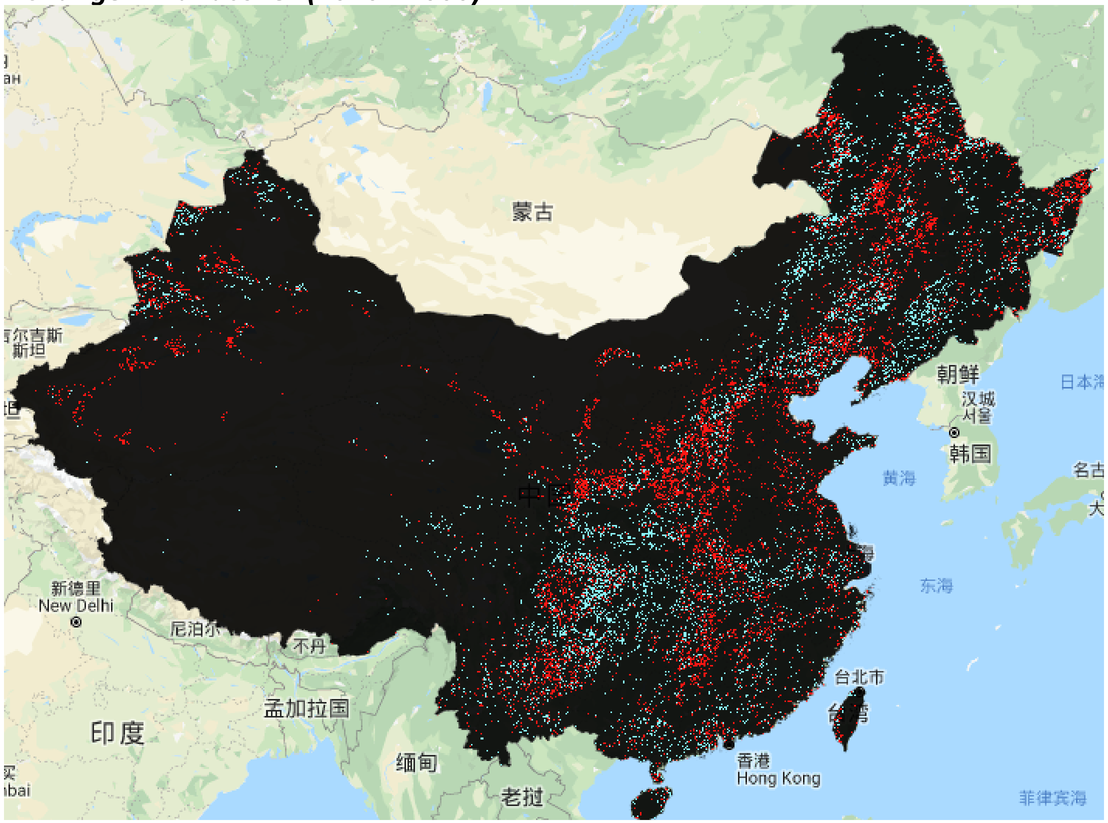
Similar to what I did in previous steps, I reclassified the map into two categories: urban and non-urban. Specifically speaking, urban area get the value 1; unclassified and undefined area get the value 10; the rest landcover get the value 2.

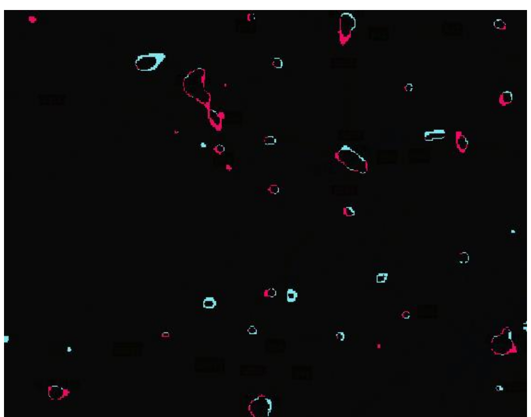
Reclassified Landcover of 2000

Reclassified Landcover of 2010

```
// Add 2000 & 2010 land cover map
Map.addLayer(landuse2010_original,{ min:0, max:17, palette:ColorsForMODIS },'landuse2010');
Map.addLayer(landuse2000_original,{ min:0, max:17, palette:ColorsForMODIS },'landuse2000')


// Reclassify the land cover types to urban & non-urban. I include class 2 and class 17 as urban area for if
// they are not concluded, the results are nearly the same.
var landuse2010 = landuse2010_original.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,10,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,10], 0, 'Land_Cover_Type_1' ).clip(CHN);
var landuse2000 = landuse2000_original.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,10,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,10], 0, 'Land_Cover_Type_1' ).clip(CHN);

// recalssified 2000 & 2010 landcover map
Map.addLayer (landuse2010,{ min:0, max:17, palette:ColorsForMODIS},'LU2010');
Map.addLayer (landuse2000,{ min:0, max:17, palette:ColorsForMODIS},'LU2000');

// Determine land use changes between 2000 and 2010
var change = landuse2010.subtract(landuse2000)

var luchange_NaturetoUrban = change.remap([0,1,8,9,-1,-8,-9],[0,1,0,0,-1,0,0],0).clip(CHN);
Map.addLayer (luchange_NaturetoUrban, {min:-1,max:1,palette:['85f7ff','000000','ff0000'],opacity:0.9}, 'landcover change');
```

Change in Landcover (2010 - 2000)





Hubei from Nightlight Data

Hubei from Landcover Data

Subtracted the reclassified landcover map of 2010 by the reclassified landcover map of 2000, the result shows on the left.

Compare to the result from nightlight map, we got more increased urban area as well as shrank areas.

To decide which result should be used. I zoomed in and found the difference:

In the result of nighttime light data, expanded and shrank areas are locate near constructed urban areas (the hollows) but in the result of landcover data, increased or decreased areas distribute more dispersed and do not seem like to be just urban areas.

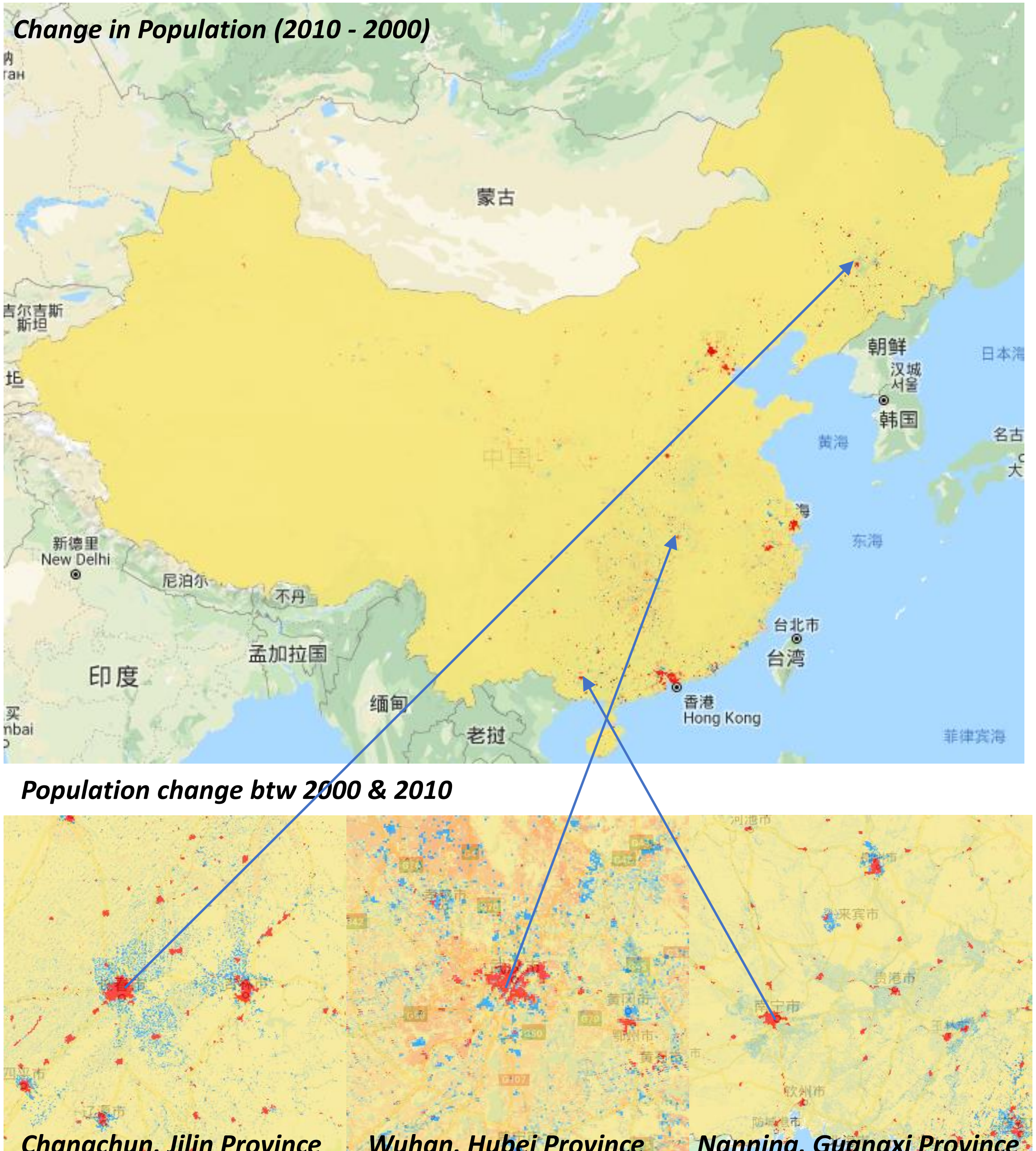
So the result from night light data is used in my latter analysis.

5. Changes of Population

Decrease in urban construction area does not necessarily means shrinking in population which is more important than urban area. So I created population map of 2000 & 2010 based on China's 5th & 6th National Population Census data (at Jiedao level), and upload in Assets for latter analysis.

Population Map of 2000

Population Map of 2010



First, subtract the raster of population in 2010 by the raster of 2000 to get the change in population of each raster unit. Then calculate the total population of each raster unit and export out as csv for analysis in ArcGIS.

Zoom in the map and we observed that capitals of provinces are grow in population while areas around them decreased (show in blue). Especially in Guangxi and Jilin Province that most part are in blue.

6. Data Wrangling

6.1 Join Shrinking Areas at Province Level

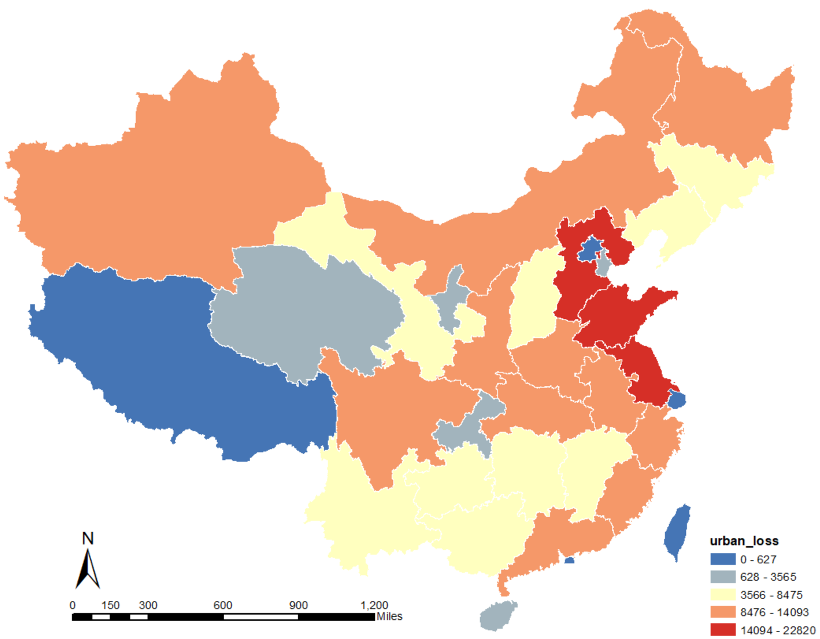
Table from Google Earth Engine contains decreased areas by Province are joined and illustrated in ArcGIS. As show in figure, the province with the most total shrinking populations are Henan, Guizhou, Jiangsu, Hubei and Guangxi Province .

NAME	ID	urban_00	urban_10	pop_00	pop_10
广东省	5	25616	39709	3665589	4888631
江苏省	15	25467	47896	3624417	3497894
河北省	9	23717	40244	3256115	3412881
浙江省	32	19023	29528	2135615	2407043
山西省	22	16956	22913	1547257	1650245
河南省	10	16888	30629	4666264	4386629
黑龙江省	11	11657	22395	1854683	1869156
辽宁省	17	11474	19949	2000850	2085669
新疆维吾	30	11336	20850	1093243	1168892
内蒙古自	18	8757	19037	1055620	1162558
陕西省	23	8745	19415	1720228	1823597
四川省	25	8416	19330	4118769	4301348
福建省	3	7951	17520	1637601	1638272
云南省	31	7643	15687	2099238	2165284
安徽省	0	6973	20514	3113351	3245817
湖北省	12	6632	15985	2740306	2617058
北京市	2	6613	7240	5676682	9385126
吉林省	14	6095	13246	1286842	1322155
天津市	27	5810	7990	1489672	6218565
湖南省	13	5423	12964	3155262	3308981
广西壮族	6	4994	12344	2252669	2139126
上海市	24	4959	5439	6335225	1158913
江西省	16	4618	11083	2187431	2207147
甘肃省	4	4038	8884	1247909	1301739
重庆市	33	3357	6922	1499560	1567750
贵州省	7	2962	8728	1858217	1686157
宁夏回族	19	2168	4673	2781470	3192816
海南省	8	1783	4321	3284493	4190104
青海省	20	1287	2983	2705501	2761950
西藏自治	28	522	934	1305424	1427490

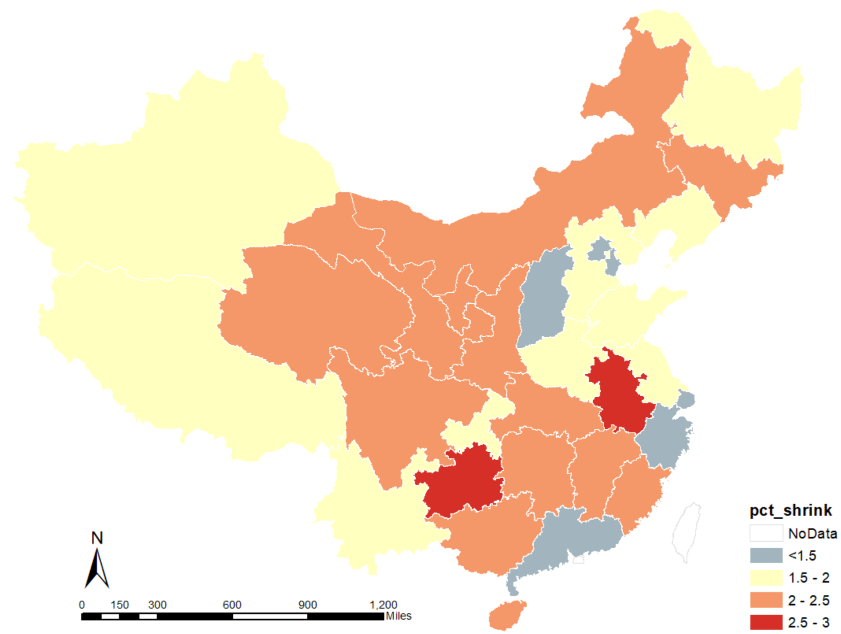
NAME	urban_loss	pop_loss	pct_shrink	degree_los
河南省	13741	-2796354	1.813655	.940073
贵州省	5766	-1720593	2.946658	.907406
江苏省	22429	-1265231	1.880708	.965091
湖北省	9353	-1232484	2.410283	.955024
广西壮族	7350	-1135430	2.471766	.949596
福建省	9569	6707	2.203496	1.00041
青海省	1696	56449	2.317793	1.020865
西藏自治	412	122066	1.789272	1.093507
黑龙江省	10738	144731	1.921163	1.007804
江西省	6465	197161	2.399957	1.009013
吉林省	7151	353128	2.173257	1.027441
宁夏回族	2505	411346	2.155443	1.147888
甘肃省	4846	538300	2.200099	1.043136
云南省	8044	660456	2.052466	1.031462
重庆市	3565	681901	2.06196	1.045473
新疆维吾	9514	756490	1.839273	1.069197
辽宁省	8475	848187	1.738626	1.042391
海南省	2538	905611	2.423444	1.275723
山西省	5957	1029879	1.351321	1.066562
陕西省	10670	1033683	2.220126	1.06009
内蒙古自	10280	1069384	2.173918	1.101304
安徽省	13541	1324667	2.941919	1.042548
湖南省	7541	1537191	2.390559	1.048718
河北省	16527	1567657	1.696842	1.048145
四川省	10914	1825795	2.296816	1.044329
山东省	22820	1984892	1.796899	1.045564
浙江省	10505	2714287	1.552226	1.127096
北京市	627	3708444	1.094813	1.653277
天津市	2180	4728893	1.375215	4.174452
上海市	480	5253912	1.096794	1.829317
广东省	14093	12230417	1.550164	1.333655

```
# Replicate the input shapefile
arcpy.Copy_management(InputShapefile,OutputShapefile)

# Join Table of Change of Urban Area / loss of population to County / Province shappefile
arcpy.JoinField_management(data = OutputShapefile, field = InputField,
                           join_table = JoinTable, join_field = JoinField)
```

```
# Retrieve attribute values (urban area and population) from the next record
urban2000 = nextRecord.getValue("urban_00")
urban2010 = nextRecord.getValue("urban_10")
pop2000 = nextRecord.getValue("pop_00")
pop2010 = nextRecord.getValue("pop_10")

# Calculate variables
PopLoss = pop2010-pop2000
DegreeLoss = pop2010 / ( pop2000 + 1)
ShrinkArea = urban2010 - urban2000
DegreeShrink = (urban2010 - urban2000) / urban2000
nextRecord.setValue(Field1,PopLoss)
nextRecord.setValue(Field2,DegreeLoss)
nextRecord.setValue(Field3,ShrinkArea)
nextRecord.setValue(Field4,DegreeShrink)
enumerationOfRecords.updateRow(nextRecord)
```



Area of Urban Loss (2010 – 2000)



Degree of Urban Loss (2010/2000)

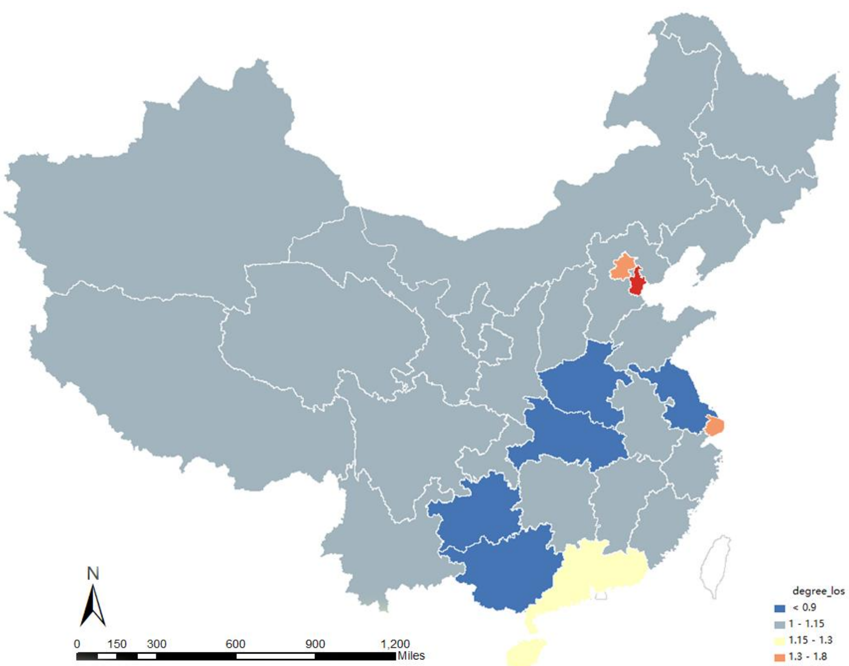
From province level, all provinces do not have urban loss during 2000 & 2010, namely urban areas in all 34 provinces are increased. Especially Hebei, Shandong and Jiangsu Province have the most increased urban areas.

When I use urban are in 2010 divided by urban area in 2000, result is little different shows that Guizhou and Jiangsu developed the most compare to their level in 2000.

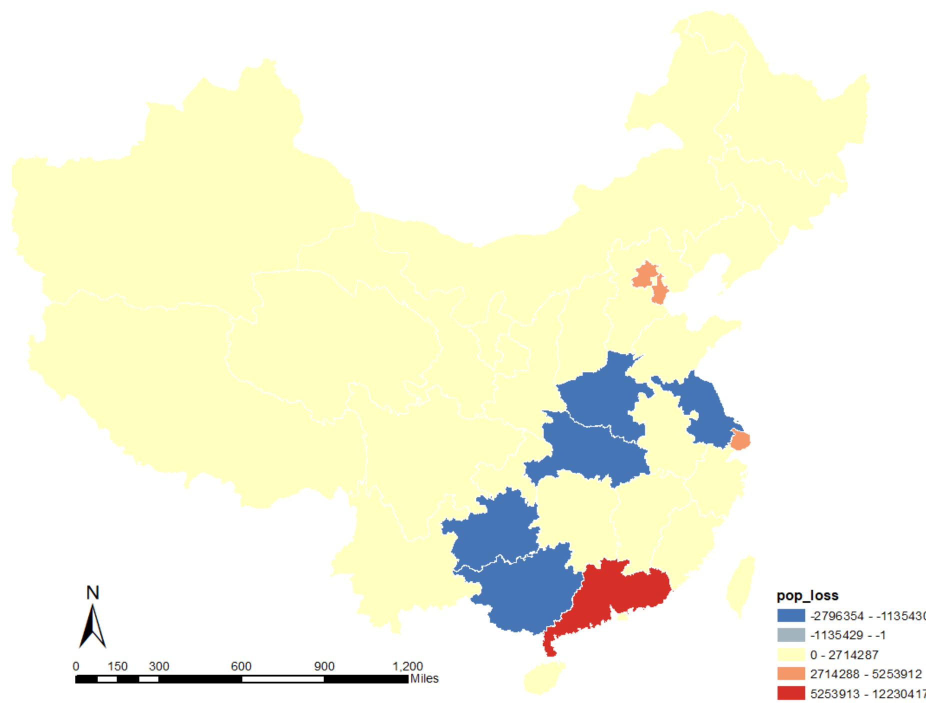
As for population aspect, five provinces have their population loss from 2000 to 2010 which are Jiangsu, Henan, Hubei, Guizhou and Guangxi Province. And the province with the most population flood in is Guangdong.

The map of pop in 2010 / pop in 2000 on the left still have these 5 provinces in blue, but only 3 provinces have their population increased more than twice: Beijing, Shanghai and Tianjin.

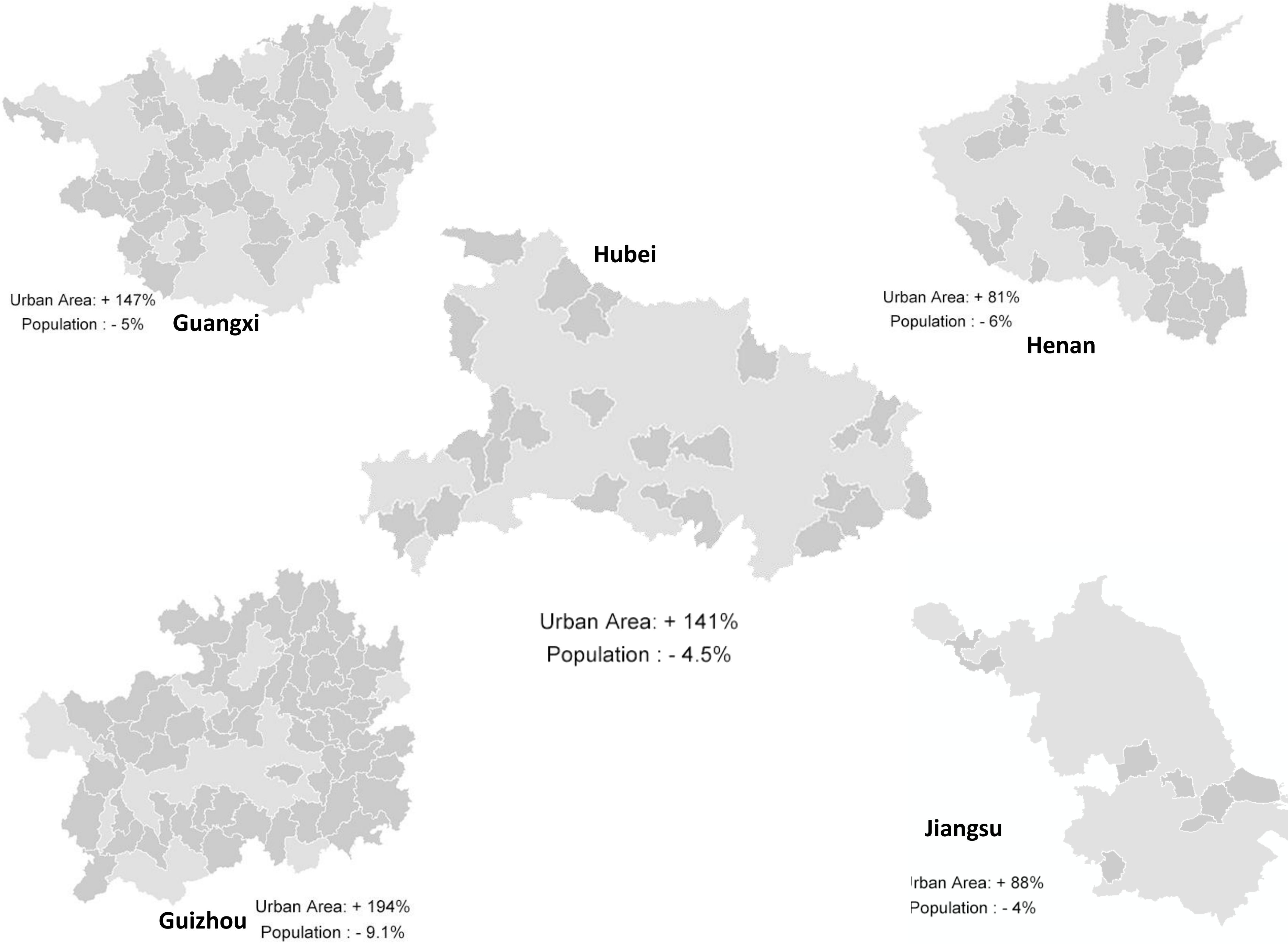
Next slide shows detailed change of urban area and population of these 5 provinces in blue.



Degree of Population Loss (2010/2000)



Population Loss (2010 – 2000)

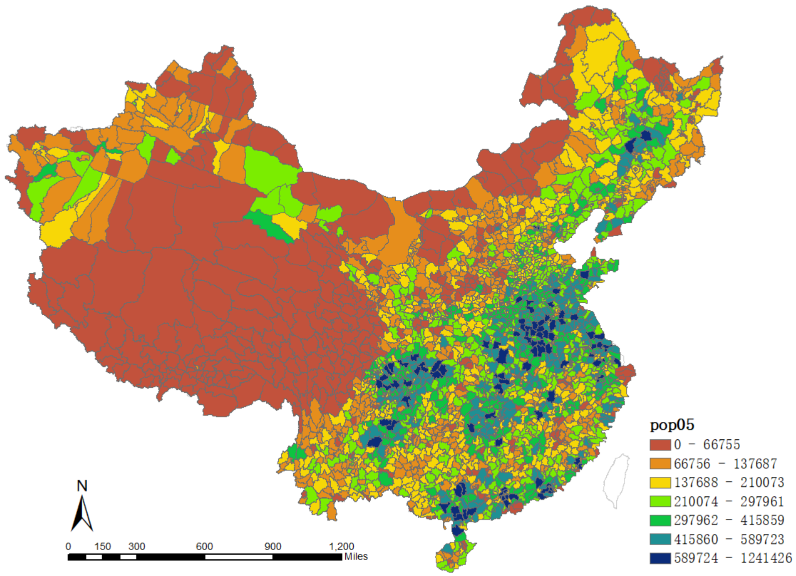

Guangxi
Urban Area: + 147%
Population : - 5%
Hubei
Urban Area: + 141%
Population : - 4.5%
Henan
Urban Area: + 81%
Population : - 6%
Guizhou
Urban Area: + 194%
Population : - 9.1%
Jiangsu
rban Area: + 88%
Population : - 4%

6.2 Join Shrinking Areas at County Level

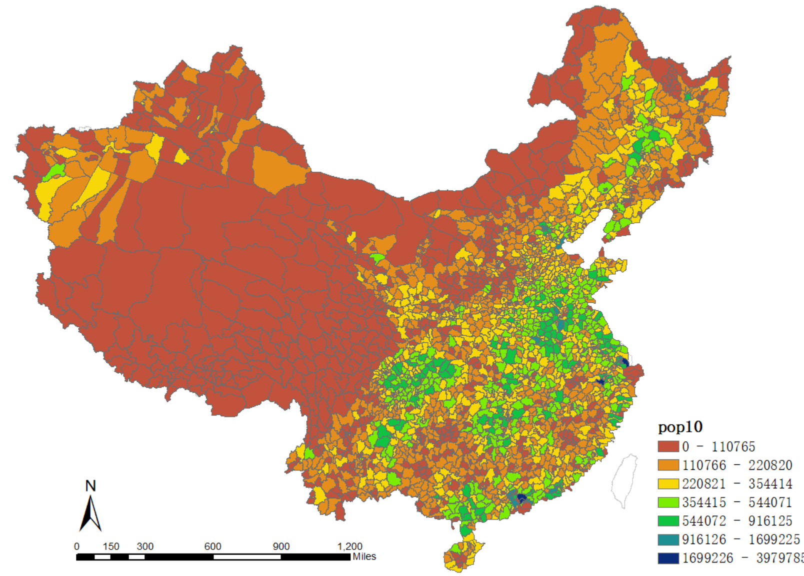
Table from Google Earth Engine contains decreased areas by Province are joined and illustrated in ArcGIS. As show in figure, the province with the most total shrinking urban areas are in Xinjiang, Shanxi.

NAME	KIND	pop_loss	urban_loss	pct_shrink	degree_los
晋江市	0137	-585081	14	1.024055	.203878
潮阳区	0137	-527330	249	1.841216	.422089
定州市	0137	-451120	153	1.859551	.216761
崇信县	0137	-414950	44	12	.105441
双阳区	0137	-348214	226	3.11215	.327017
阿克塞哈萨	0137	-337264	4	1.210526	.020530
固始县	0137	-315827	97	2.564516	.596666
吴中区	0137	-298273	491	1.544346	.605849
天宁区	0137	-279911	0	1	.393142
钟楼区	0137	-251270	0	1	.419403
武进区	0137	-242641	166	1.176972	.640549
闽侯县	0137	-218140	276	1.890323	.565980
花都区	0137	-209945	148	1.221557	.702205
高港区	0137	-205290	55	1.292553	.404844
南安市	0137	-202923	642	1.958209	.709158
两当县	0137	-194640	29	0	.123930
易县	0137	-192292	39	1.6	.253623
肃北蒙古族	0137	-188241	7	0	.036894
高州市	0137	-185980	134	3.196721	.767605
上蔡县	0137	-184585	69	2.604651	.724011
临泽县	0137	-179050	39	2.857143	.272323
郸城县	0137	-175838	34	1.666667	.729427
瓜州县	0137	-175344	83	2.509091	.286218
长沙县	0137	-172254	422	2.465278	.644324
宣州区	0137	-171220	279	4.244186	.699377
岳麓区	0137	-171143	0	1	.556931
八步区	0137	-165106	123	2.556962	.625181
监利县	0137	-164873	193	4.784314	.777508
荆州区	0137	-164727	76	1.938272	.392142
沧浪区	0137	-163253	0	1	.280802
化州市	0137	-161786	170	4.035714	.766332

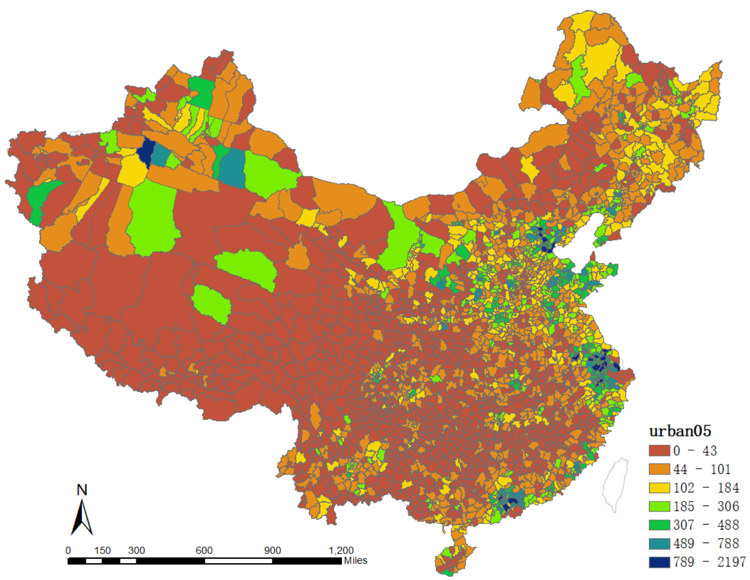
Counties with population loss btw 2000 & 2010



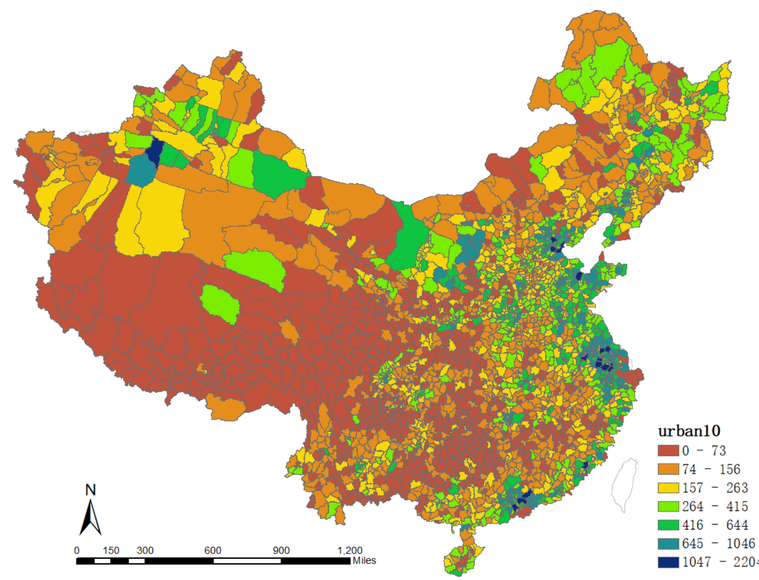
Population in 2000 (county level)



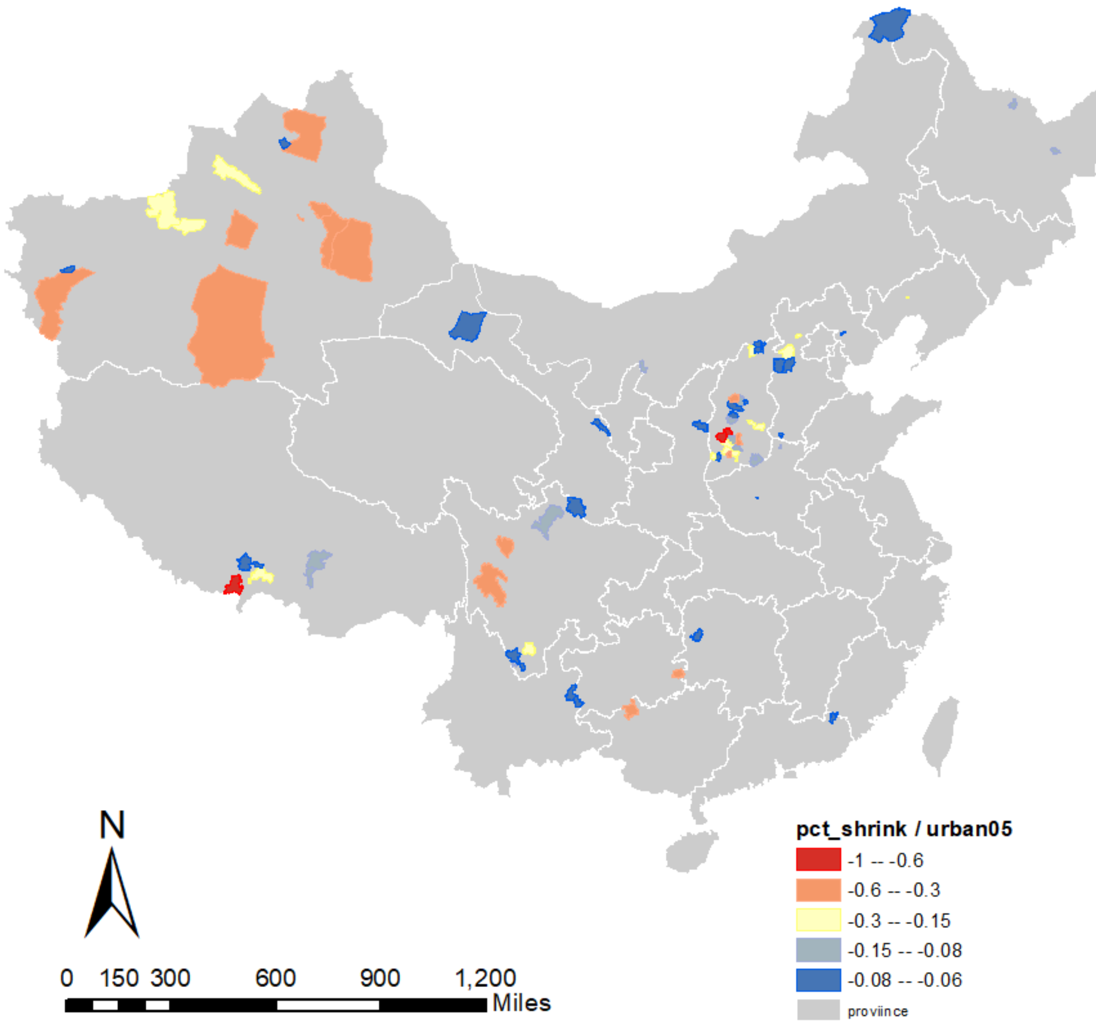
Population in 2010(county level)



Constructed area in 2000 (count by raster units)



Constructed area in 2010(count by raster units)

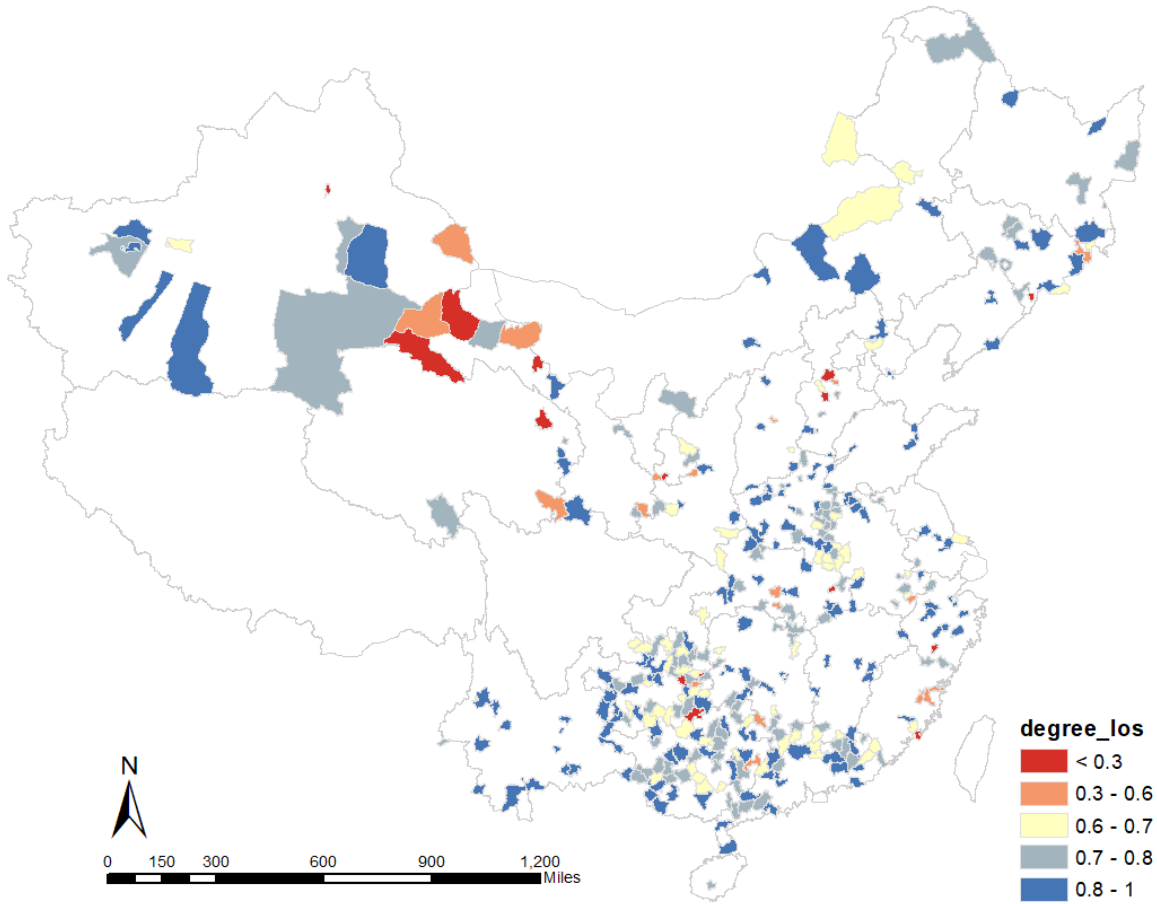


Shrinking Cities (by urban area) 2000 & 2010

6.3 Select shrinking cities by area

First, I select counties or cities with urban area shrank in 2010 compare with 2000. As shows on the left.

68 counties are of this type and mostly located in Xinjiang and Shanxi Province.



Shrinking Cities (by population) 2000 & 2010

6.4 Select shrinking cities by population

333 counties and 77 cities have their population in 2010 less than in 2000.
But more than half of them (207) have their value of (population in 2010/population in 2000) in the interval of 0.8 – 1.

Counties with value less than 0.5 are mostly located in Gansu.

7. Discussion

Though from province level, whole nation is rapidly urbanizing, if we zoom in to county level, some shrinking townships are revealed. Compare with number of townships with population loss, we have less counties with shrinking urban area between 2000 and 2010.

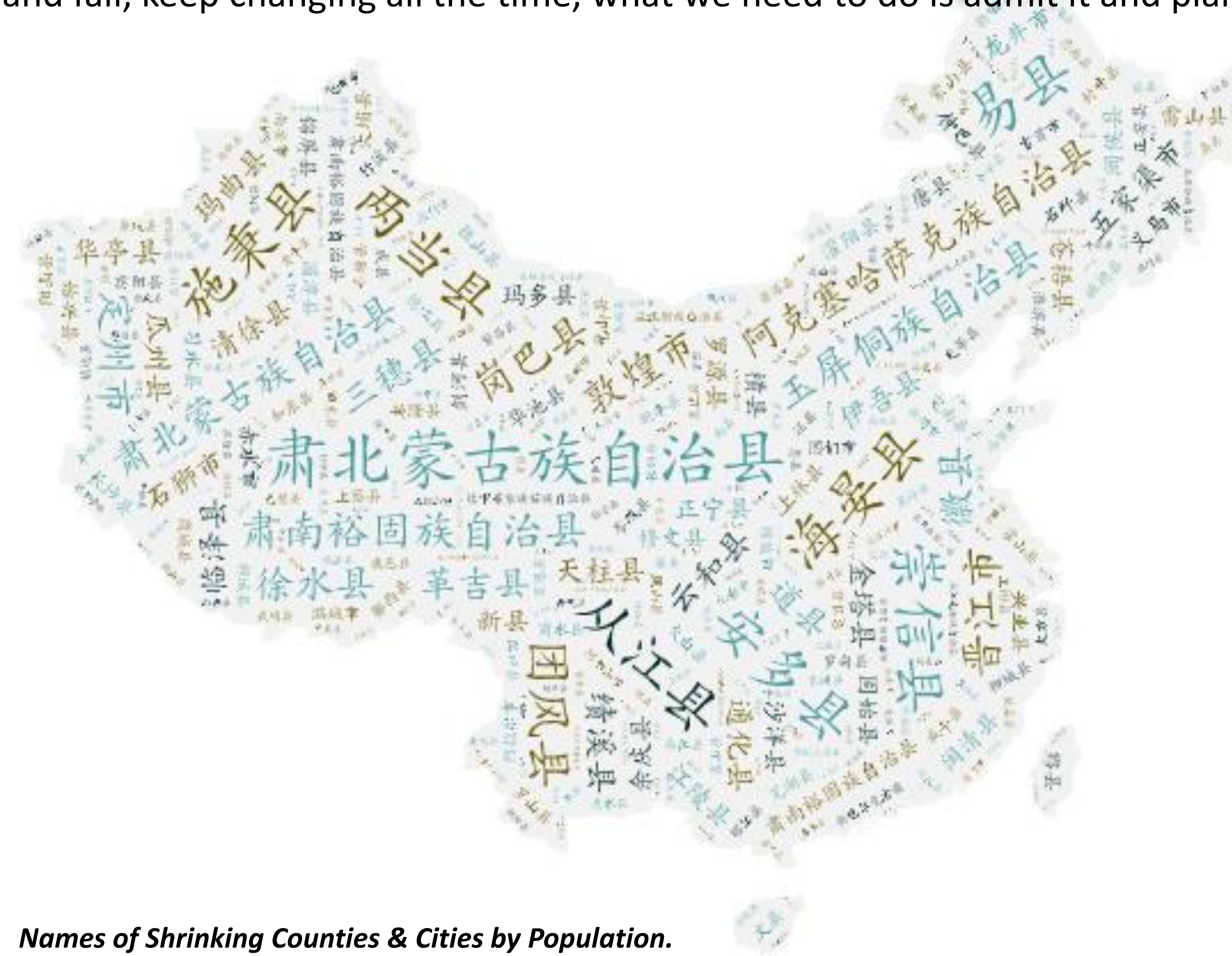
Shrinking counties in Xinjiang and Shanxi Province are with the most loss of urban area and most of them had a prosperous past due to mineral resources and oriented heavy industries. But with single industry structure and resource depletion, shrinking is inevitable.

Names of Shrinking Counties & Cities by Shrinkage of Urban Area.

7. Discussion

From the maps, we observed shrinking phenomenon is existed in China. Nearly 1/3 territories of China lost its population in this decade. Among these shrinking townships, we observed 1 provincial capital: Urumqi (Xinjiang); 77 cities and 332 counties. Though major cities in China are still growing, small cities need to face the challenge to manage decline, especially persistent and accompanied with aging population, and do not stick to the past prosperity.

Cities rise and fall, keep changing all the time, what we need to do is admit it and plan for it.



Names of Shrinking Counties & Cities by Population.

Limitations

1. Light threshold chose to determine urban area may well present the urban area on nation's level, but for provinces even county level, the same threshold may result in different degree of errors.
2. Due to time & data limit, I do not consider the change of county boundaries in the past decade but created raster of population based on census data at Jiedao level.
3. Landcover data have two unclassified land types which may contain constructed, suburban areas. To be conservative, I do not consider them as urban area.
4. I did not consider Hongkong, Macao and Taiwan for lack of data.
5. The population data is only available for 2000 & 2010. And another 10 years passed, how about these shrinking counties? Do they still grow in number or some of them have recovered? We wouldn't know until the 7th National Census data released next year.
6. Results at some regions do not reflect the reality like Sichuan, Chongqing have no population loss counties and Northeastern region has very few cities showing urban loss.

Literatures Cited

Rus in urbe redux: https://www.economist.com/international/2015/05/30/rus-in-urbe-redux

Shrinking cities in a rapidly urbanizing China. Ying Long, Kang Wu, Environment and Planning A, 2016, Vol. 48(2) 220–222.

The young and lonely hearts of China's shrinking cities, National Geography: https://www.nationalgeographic.com/culture/2019/01/young-lonely-hearts-millennial-northeast-china-shrinking-cities/

2019.03 EDGY: Data Shows One Third of Chinese Cities are Shrinking: https://edgy.app/chinese-urban-areas-shrinking?pfrom=home

Smaller Chinese cities are dwindling despite the country's attempt at urbanization, Global Times Published: 2018/3/27

GEE Codes Link:
https://code.earthengine.google.com/5841ee61164b7624014c4368ea40dc69

```javascript
//Import the China's Boundary / Provinces / Counties from Assets.
var CHN = ee.FeatureCollection('users/derrickshuxx/china');
var province = ee.FeatureCollection('users/derrickshuxx/province');
var county = ee.FeatureCollection('users/derrickshuxx/county84');
Map.addLayer(CHN);
Map.addLayer(province);
Map.addLayer(county);


// Part1. Use nighttime light data to determine the change in urban areas.
// Import the nighttime lights image and clip to China's boundary
var world_nightlights = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS');
print('world nighttime lights', world_nightlights);
var world2000 = ee.Image('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS/F152000').select('stable_lights')
var light2000 = world2000.clip(CHN);
var world2010 = ee.Image('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS/F182010').select('stable_lights')
var light2010 = world2010.clip(CHN);

// Add 2000 & 2010 night light map
Map.addLayer(light2000,{min:0,max:63,palette:['000000','555000','fff000'],opacity:0.9},'stable light 2000')
Map.addLayer(light2010,{min:0,max:63,palette:['000000','555000','fff000'],opacity:0.9},'stable light 2010')


// To determine the light threshold, I use ArcGIS to calculate the urban area under 63 thresholds and
// compare them to official data from China's Urban Yearbook of 2001 & 2011.
// Based on analysis in ArcGIS, I define the areas with nightlight values larger than 15
// as urban areas in 2000 and areas with nightlight values larger than 30 as urban areas in 2010.
// Then reclassify the urban areas in 2000 and 2010 respectively (Urban area:1 & non-urban area :0)

var urban2000 = light2000.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], 1, 'stable_lights' ).clip(CHN);
var urban2010 = light2010.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30],
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], 1, 'stable_lights' ).clip(CHN);
Map.addLayer(urban2000,{min:0,max:1,palette:['000000','ff1900'],opacity:0.9},'urban2000')
Map.addLayer(urban2010,{min:0,max:1,palette:['000000','ff1900'],opacity:0.9},'urban2010')

// Subtract reclassified raster of 2000 by reclassified raster of 2010 to get the three types of change
// in urban areas: 1 - developed; 0 - unchanged; -1 - shrank.
var change_urban = urban2010.subtract(urban2000)
Map.addLayer (change_urban, {min:-1,max:1,palette:['85f7ff','000000','ff0066'],opacity:0.9}, 'shrink' );

// Select shrinking rasters
var shrink_urban = change_urban.gt(-1)

// Aggregate shrinking area into provinces
// get the sum of raster number of decreased areas of each province
var province_shrink = shrink_urban.reduceRegions({
  collection: province,
  reducer:ee.Reducer.sum(),
  scale:1000
})
print(ee.Feature(province_shrink.first()),'province_shrink');

// get the sum of raster number of decreased areas of each county
var county_shrink = shrink_urban.reduceRegions({
  collection: county,
  reducer:ee.Reducer.sum(),
  scale:1000
})
print(ee.Feature(county_shrink.first()),'county_shrink');

// Make histogram of the shrinking areas by province
var charts_pr = Chart.feature.histogram(province_shrink, 'sum',50)
var charts_pr = charts_pr.setOptions({ title:'Number of shrinking rasters by Province',
colors : ['ffc400'],
hAxis: {title:'Number of Rasters'},
vAxis: {title:'Number of Provinces'}
});
print(charts_pr)

// Make histogram of the shrinking areas by county
var charts_ct = Chart.feature.histogram(county_shrink, 'sum',50)
var charts_ct = charts_ct.setOptions({ title:'Number of shrinking rasters by County',
colors : ['ffc400'],
hAxis: {title:'Number of Rasters'},
vAxis: {title:'Number of Counties'}
});
print(charts_ct)
```

```
// Part2 --- Use Landcover data to determine the change in urban areas.
// To be conservative, I use landcover data to determine the urban areas just in case.

// Import the land cover image for 2000 and 2010
var landuse2010_original = ee.Image('MODIS/051/MCD12Q1/2010_01_01').select('Land_Cover_Type_1').clip(CHN);
var landuse2000_original = ee.Image('MODIS/051/MCD12Q1/2001_01_01').select('Land_Cover_Type_1').clip(CHN);

// add colors based on class example
var palette_lu = ['00b7ff',
'33510d',
'b8b6b4',
'3b9c4b',
'20f551',
'32783e',
'ff6100',
'c3aa69',
'b76031',
'd9903d',
'a9c952',
'111149',
'ebcc3f',
'f0000c',
'33280d',
'd7cdcc',
'f7e084',
'6f6f6f'
]

// Add 2000 & 2010 land cover map
Map.addLayer(landuse2010_original,{ min:0, max:17, palette:palette_lu },'landuse2010');
Map.addLayer(landuse2000_original,{ min:0, max:17, palette:palette_lu },'landuse2000')


// Reclassify the land cover types to urban & non-urban. I include class 2 and class 17 as urban area for if
// they are not concluded, the results are nearly the same.
var landuse2010 = landuse2010_original.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,10,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,10], 0, 'Land_Cover_Type_1' ).clip(CHN);
var landuse2000 = landuse2000_original.remap( [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17],
[1,10,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,10], 0, 'Land_Cover_Type_1' ).clip(CHN);

// recalssified 2000 & 2010 landcover map
Map.addLayer (landuse2010,{ min:0, max:17, palette:palette_lu},'LU2010');
Map.addLayer (landuse2000,{ min:0, max:17, palette:palette_lu},'LU2000');

// Determine land use changes between 2000 and 2010
var change = landuse2010.subtract(landuse2000)

var landchange = change.remap([0,1,8,9,-1,-8,-9],[0,1,0,0,-1,0,0],0).clip(CHN);
Map.addLayer (landchange, {min:-1,max:1,palette:['85f7ff','000000','ff0000'],opacity:0.9}, 'landcover change');
```

```
// Part3. Use Population data to determine the change in population.
// Import Population Data of 2000 and 2010
var pop05 = ee.Image("users/derrickshuxx/Population2005");
var pop10 = ee.Image("users/derrickshuxx/cnpop2010");
Map.addLayer(pop05,{min:0,max:1500,palette:['0095ff','f7e779','ff0400'],opacity:0.9},'population 2005')
Map.addLayer(pop10,{min:0,max:1200,palette:['0095ff','f7e779','ff0400'],opacity:0.9},'population 2010')

// get the sum of population of each county & province
var county_05 = pop05.reduceRegions({
  collection: county,
  reducer:ee.Reducer.sum(),
  scale:1000
})

var county_10 = pop10.reduceRegions({
  collection: county,
  reducer:ee.Reducer.sum(),
  scale:1000
})

var province_05 = pop05.reduceRegions({
  collection: province,
  reducer:ee.Reducer.sum(),
  scale:1000
})

var province_10 = pop10.reduceRegions({
  collection: province,
  reducer:ee.Reducer.sum(),
  scale:1000
})

// Part4. Join population and Urban features of counties and provinces
var innerJoin = ee.Join.inner();
var filter1 = ee.Filter.equals({
leftField: 'NAMES',
rightField: 'NAMES'
})
var innerJoined_cty = innerJoin.apply(county_shrink, county_05, county_10,filter1);
var innerJoined_prv = innerJoin.apply(province_shrink, province_05, province_10,filter1);

// Get the map of population change
var pop_change = pop10.subtract(pop05)
Map.addLayer(pop_change,{min:-3000,max:3000,palette:['0095ff','f7e779','ff0400'],opacity:0.9},'pop_change')


// Export the results out as Geotiff / .csv files

Export.image.toDrive({
  image: change_urban,
  description: 'change of urban area',
  scale: 10000,
  region: CHN,
  fileFormat: 'GeoTIFF',
  formatOptions: {
    cloudOptimized: true
  }
});


Export.image.toDrive({
  image: pop_change,
  description: 'population change',
  scale: 10000,
  region: CHN,
  fileFormat: 'GeoTIFF',
  formatOptions: {
    cloudOptimized: true
  }
});

Export.table.toDrive({
collection: innerJoined_cty,
description:'County_change',
fileFormat: 'CSV'
});

Export.table.toDrive({
collection: innerJoined_prv,
description:'Province_change',
fileFormat: 'CSV'
});
```

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> """
THIS SCRIPT JOINS THE ATTRIBUTE TABLE OF ONE FEATURE LAYER TO THAT OF ANOTHER
AND SAVES THE RESULT AS A NEW SHAPEFILE

To create an ArcToolbox tool with which to execute this script, do the following.
1   In  ArcMap > Catalog > Toolboxes > My Toolboxes, either select an existing toolbox
    or right-click on My Toolboxes and use New > Toolbox to create (then rename) a new one.
2   Drag (or use ArcToolbox > Add Toolbox to add) this toolbox to ArcToolbox.
3   Right-click on the toolbox in ArcToolbox, and use Add > Script to open a dialog box.
4   In this Add Script dialog box, use Label to name the tool being created, and press Next.
5   In a new dialog box, browse to the .py file to be invoked by this tool, and press Next.
6   In the next dialog box, specify the following inputs (using dropdown menus wherever possible)
    before pressing OK or Finish.
        DISPLAY NAME            DATA TYPE           PROPERTY>DIRECTION>VALUE    PROPERTY>OBTAINEDFROM>VALUE
        Input Shapefile         Shapefile           Input
        Urban/pop table         Data Table          Input
        Join Field 1            String              Input
        Join Field 2            String              Input
        Output Shapefile        Shapefile           Output
7   To later revise any of this, right-click to the tool's name and select Properties.
"""

# Import necessary modules
import sys, os, string, math, arcpy, traceback, numpy

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

try:
    # Request user input of data type
    InputShapefile = arcpy.GetParameterAsText(0)
    JoinTable      = arcpy.GetParameterAsText(1)
    InputField     = arcpy.GetParameterAsText(2)
    JoinField      = arcpy.GetParameterAsText(3)
    OutputShapefile = arcpy.GetParameterAsText(4)

    # Replicate the input shapefile
    arcpy.Copy_management(InputShapefile,OutputShapefile)

    # Join Table of Change of Urban Area / loss of population to County / Province shappefile
    arcpy.JoinField_management(data = OutputShapefile, field = InputField,
                               join_table = JoinTable, join_field = JoinField)

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

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> """
THIS SCRIPT JOINS THE ATTRIBUTE TABLE OF ONE FEATURE LAYER TO THAT OF ANOTHER
AND SAVES THE RESULT AS A NEW SHAPEFILE

To create an ArcToolbox tool with which to execute this script, do the following.
1   In  ArcMap > Catalog > Toolboxes > My Toolboxes, either select an existing toolbox
    or right-click on My Toolboxes and use New > Toolbox to create (then rename) a new one.
2   Drag (or use ArcToolbox > Add Toolbox to add) this toolbox to ArcToolbox.
3   Right-click on the toolbox in ArcToolbox, and use Add > Script to open a dialog box.
4   In this Add Script dialog box, use Label to name the tool being created, and press Next.
5   In a new dialog box, browse to the .py file to be invoked by this tool, and press Next.
6   In the next dialog box, specify the following inputs (using dropdown menus wherever possible)
    before pressing OK or Finish.
        DISPLAY NAME            DATA TYPE           PROPERTY>DIRECTION>VALUE    PROPERTY>OBTAINEDFROM>VALUE
        Input Shapefile         Shapefile           Input
        Urban/pop table         Table               Input
        Join Field 1            String              Input
        Join Field 2            String              Input
        Output Shapefile        Shapefile           Output
        Output Table            Table               Output

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

# Import necessary modules
import sys, os, string, math, arcpy, traceback, numpy

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

try:
    # Request user input of data type
    InputShapefile = arcpy.GetParameterAsText(0)
    JoinTable      = arcpy.GetParameterAsText(1)
    InputField     = arcpy.GetParameterAsText(2)
    JoinField      = arcpy.GetParameterAsText(3)
    OutputShapefile = arcpy.GetParameterAsText(4)
    OutputTable     = arcpy.GetParameterAsText(5)

    # Replicate the input shapefile
    arcpy.Copy_management(InputShapefile,OutputShapefile)

    # Join Table of Change of Urban Area / loss of population to County / Province shappefile
    arcpy.JoinField_management(data = OutputShapefile, field = InputField,
                               join_table = JoinTable, join_field = JoinField)
    # Join the urban & population data of each province and county by their NAME
    # calculate population loss / degree of population loss / shrank urban area / degree of urban shrink
    Field1 = "pop_loss"
    arcpy.AddMessage("The name of the field to be added is" + Field1 + "\n")

    Field2 = "degree_loss"
    arcpy.AddMessage("The name of the field to be added is" + Field2 + "\n")

    Field3 = "urban_loss"
    arcpy.AddMessage("The name of the field to be added is" + Field3 + "\n")
```

```
    Field4 = "pct_shrink"
    arcpy.AddMessage("The name of the field to be added is" + Field4 + "\n")

    # Add new fields to the output shapefile
    arcpy.AddField_management(OutputShapefile, Field1, "LONG INTEGER", 10)
    arcpy.AddField_management(OutputShapefile, Field2, "DOUBLE", 20, 5)
    arcpy.AddField_management(OutputShapefile, Field3, "LONG INTEGER", 10)
    arcpy.AddField_management(OutputShapefile, Field4, "DOUBLE", 20, 5)

    # Create an enumeration of updatable records from the shapefile's attribute table
    enumerationOfRecords = arcpy.UpdateCursor(OutputShapefile)

    # Loop through records, calculating each population loss / degree of population loss / shrank urban area
    # / degree of urban shrink
    nextRecords = arcpy.UpdateCursor(OutputShapefile)
    for nextRecord in enumerationOfRecords:
            nextRecord.setValue(Field8,urban10)
            enumerationOfRecords.updateRow(nextRecord)
    # Delete row and update cursor objects to avoid locking attribute table
    del nextRecord
    del enumerationOfRecords

    # Retrieve attribute values (urban area and population) from the next record's Shape field
    urban2000 = nextRecord.getValue("urban_00")
    urban2010 = nextRecord.getValue("urban_10")
    pop2000 = nextRecord.getValue("pop_00")
    pop2010 = nextRecord.getValue("pop_10")

    # Calculate variables
    PopLoss = pop2010-pop2000
    DegreeLoss = pop2010 / ( pop2000 + 1)
    ShrinkArea = urban2010 - urban2000
    DegreeShrink = (urban2010 - urban2000) / urban2000
    nextRecord.setValue(Field1,PopLoss)
    nextRecord.setValue(Field2,DegreeLoss)
    nextRecord.setValue(Field3,ShrinkArea)
    nextRecord.setValue(Field4,DegreeShrink)
    enumerationOfRecords.updateRow(nextRecord)

    # Delete row and update cursor objects to avoid locking attribute table
    del nextRecord
    del enumerationOfRecords

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