Night-time Lights:

Trend of Human Activities Across the World

Yixuan HU, MUSA candidate in PennDesign

Advisor: Prof. Dana Tomlin

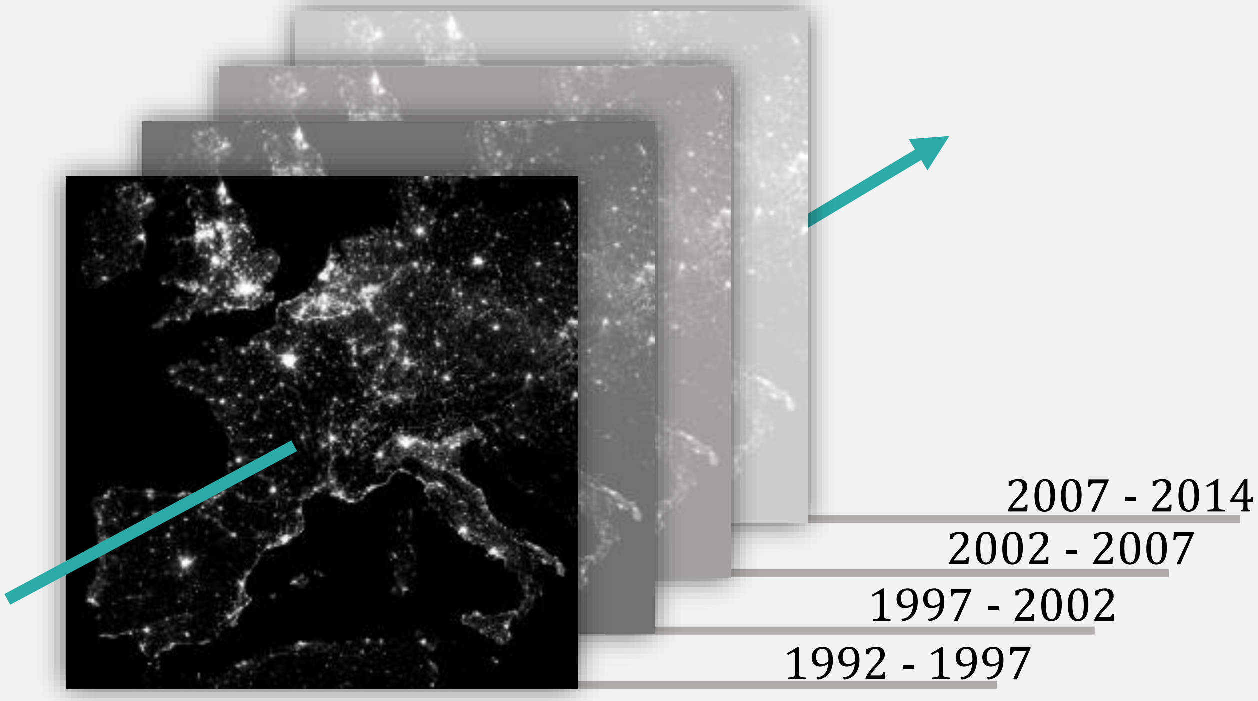
Introduction

Human activities have transformed the surface of the globe. **Night-time light(DTL)** is the footprint of human activities we can see through satellite imagery, and can help us explore the impact and intensity of human activities at a global scale.

Dating back to 1992, the night-time light data provided by Defense Meteorological Satellite Program (DMSP) are the longest-running time series of global data of this kind and the DTL data is the best source to study the **trend of human activities** in the whole **world**.

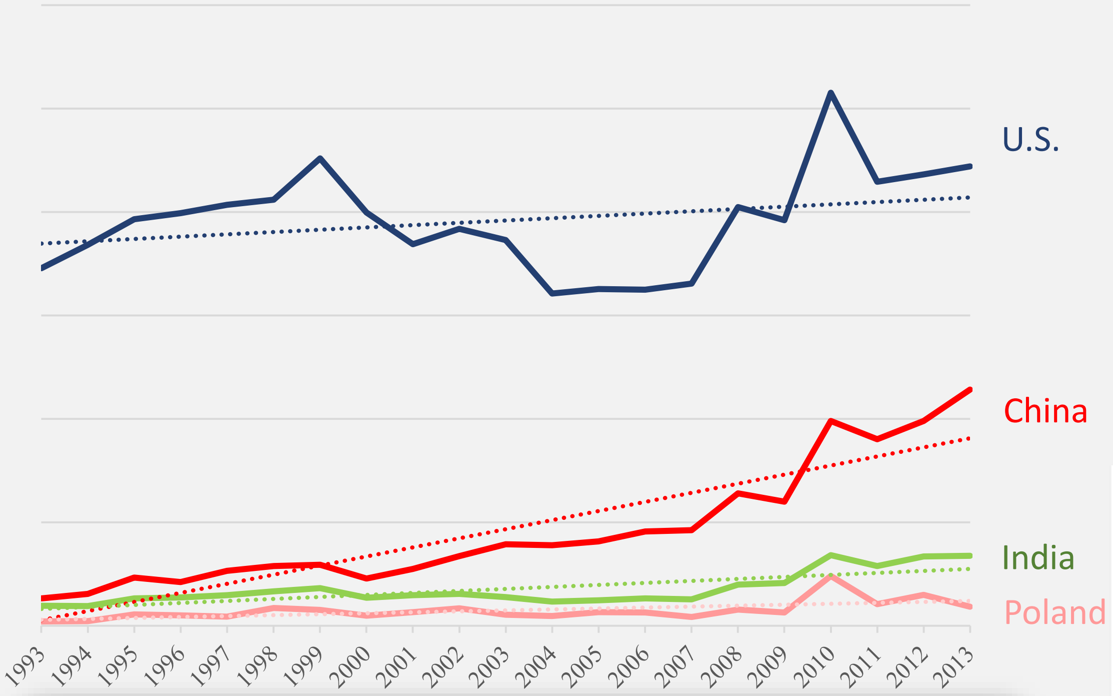
Google Earth Engine, with its immense storage and computing power makes large scale analysis in everywhere possible.

Framework



- Explore the global trend in **night-time lights** from **1992** to **2014** and **4 sub-periods** in regions in the world
- Identify areas with certain characteristics of growth:
  - the fastest 5% growing areas
  - the “texture” of growth
  - aggregated growth area
- Visualize the night-time lights growth in states in the U.S. with:
  - sum of growth rate
  - sum of growth rate normalized by area

Lit Area Across the Globe



I identified areas with a digital number larger than 31 (63 is the maximum DN) as *lit area*, calculate the sum of lit area within regions, and export it as a table.

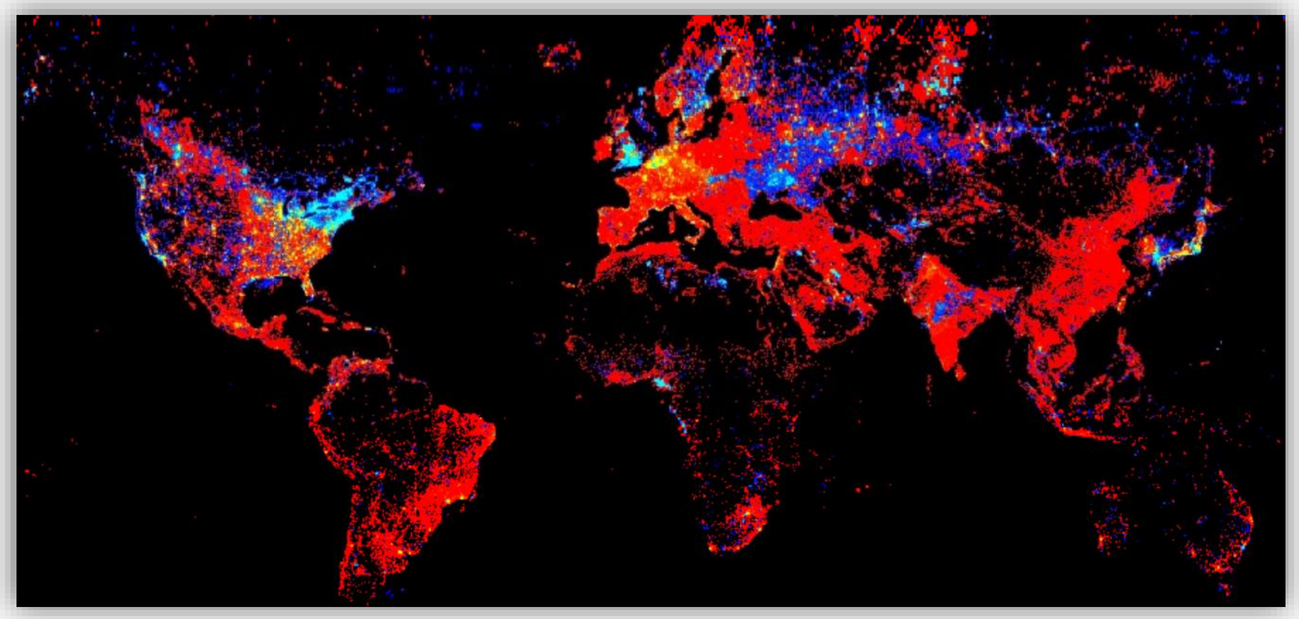
From the upper-left plot, though all four countries experienced growth in night-time light intensity, we could see the curves are different at different time intervals.


```
function AreaLit(i){
  var thold = blank.where(i.gt(31),1); //Set the threshold of lit - 31
  var alit = thold.mask(thold);
  alit = ee.Image.pixelArea().mask(alit);
  alit = alit.divide(1000000);
  alit = alit.set('index',i.get('system:index'));
  return alit;
  }

function tabulate(i){
  return Regions.map(function (f){
    var r = i.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort:true,
      maxPixels:1e9});
    return ee.Feature(null, {
      name: f.get('name'),
      area: r.get('area'),
      index: i.get('index')});
  });}

var AreaLit_Results = World_all.map(AreaLit); //A ImageCollection containing the lit area
var AreaLit_ResultsTable = AreaLit_Results.map(tabulate).flatten(); //A table with lit area info in every countries
Export.table.toDrive(AreaLit_ResultsTable,'DMSP_results','DMSP results','results','csv');
```

Night-time Light Change

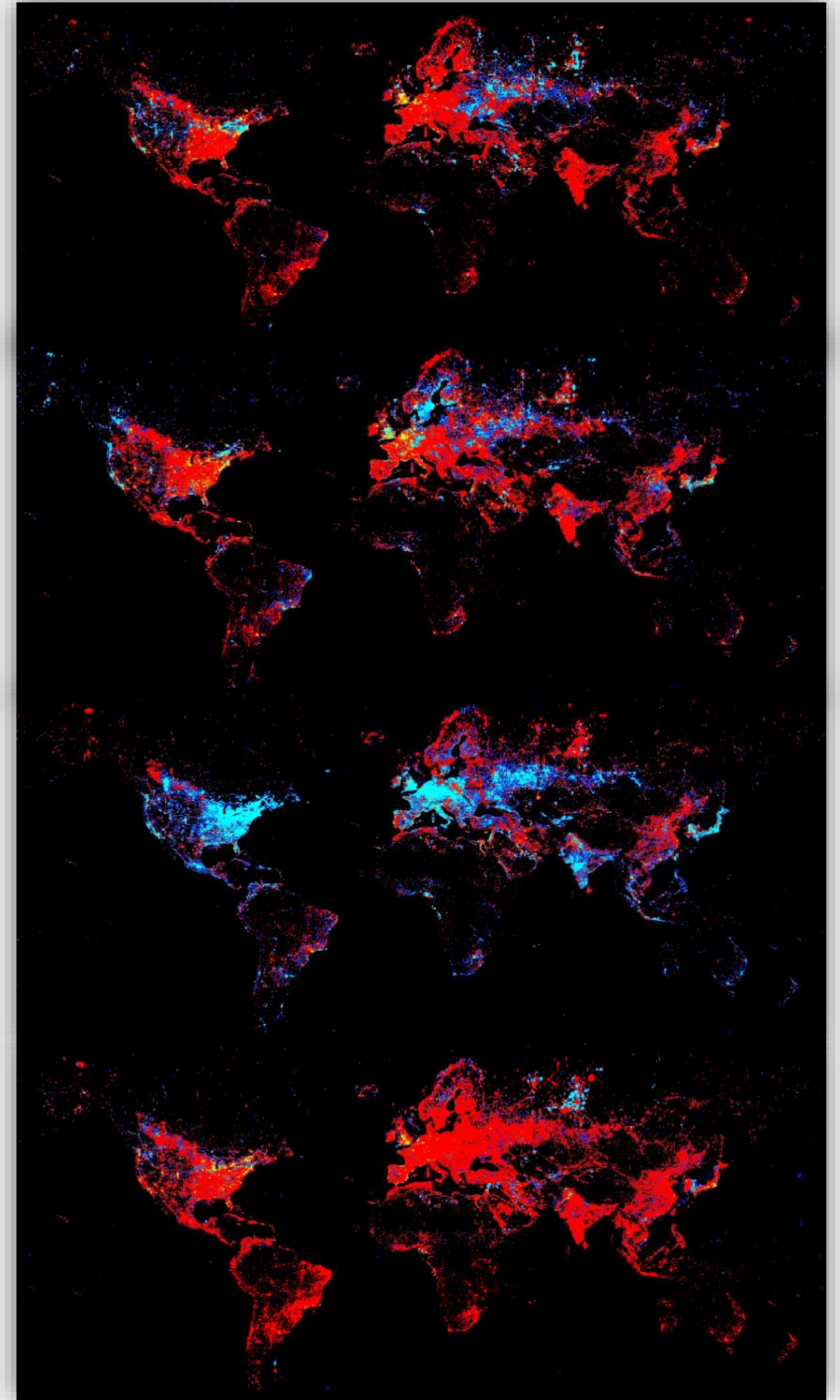


1992 – 2014

I used linear regression to calculate the change rate of night-time lights in 4 sub-periods, and assigned growth rate, brightness (intercept), and decline rate in red, green, and blue bands, respectively.

```
var GetYear = function(y){
  return ee.Date.fromYMD({
    day: 1, month: 1, year: y});
};
```

```
var func_LinearFit = function(year1, year2){
  var SYear = GetYear(year1);
  var EYear = GetYear(year2);
  var world_TimeBand = World_all.filterDate(SYear, EYear).map(createTimeBand_fromLinearRegression);
  var linearFit = world_TimeBand.select(['system:time_start','stable_lights'])
    .reduce(ee.Reducer.linearFit());
  return linearFit;
};
```



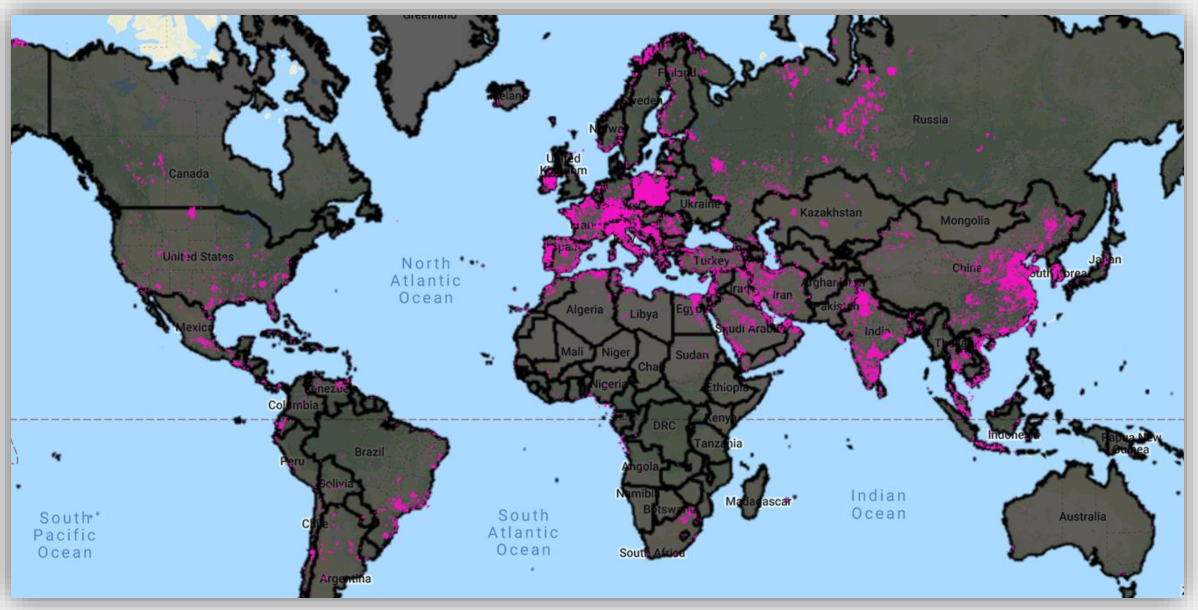
1992 – 1997

1997 – 2002

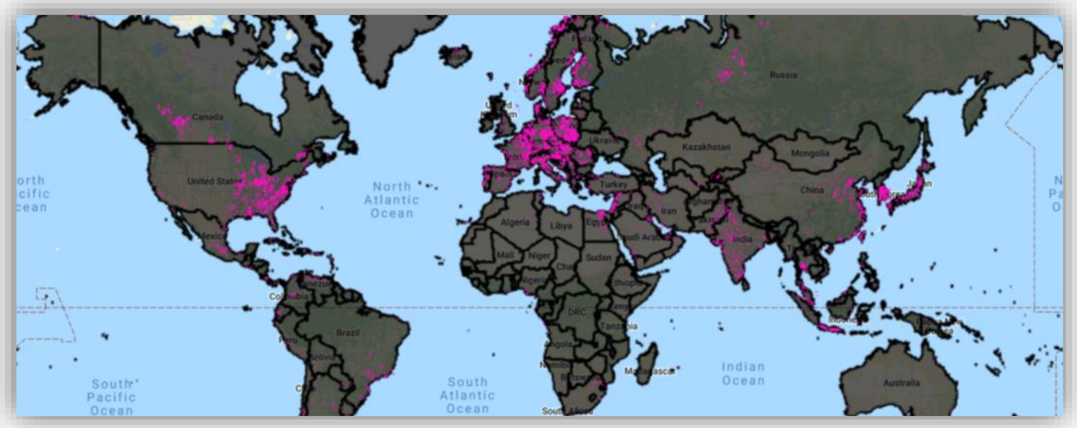
2002 – 2007

2007 – 2014

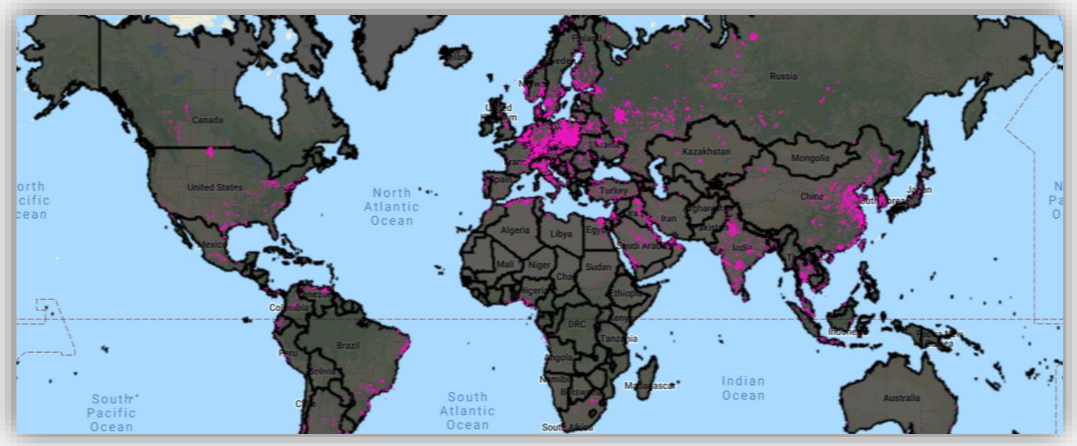
Fastest Growing Areas



1992 – 2014



1992 – 1997

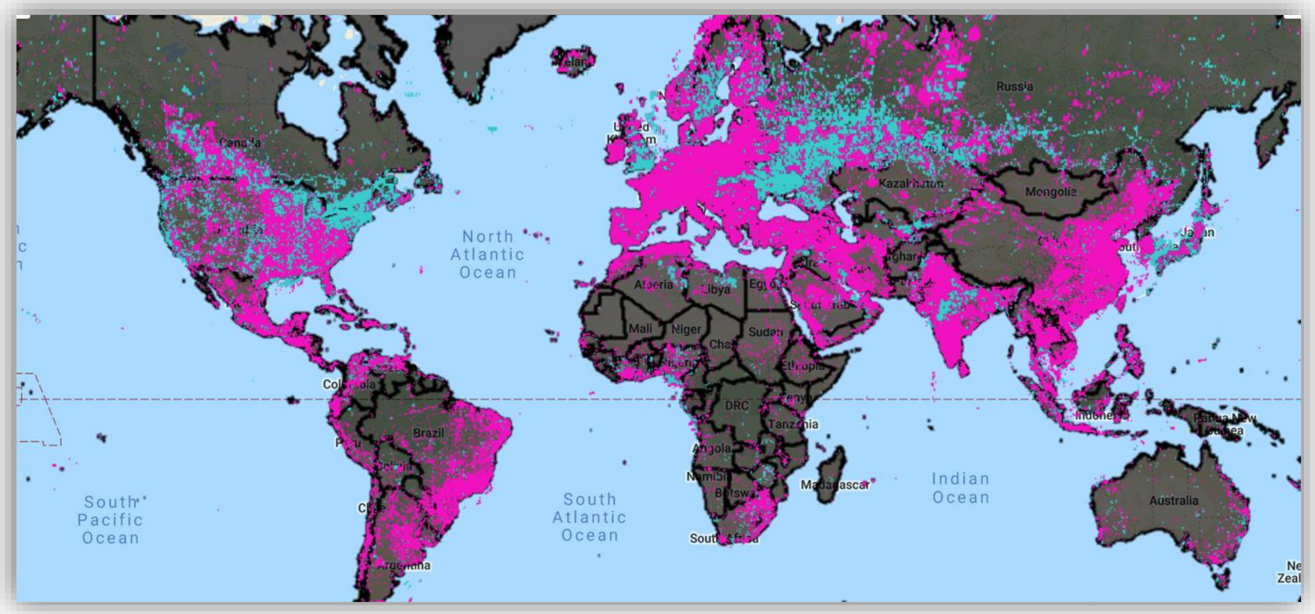


2007 – 2014

I identified the fastest growing 5% of areas across three time intervals and found that areas in North America was among the fastest between 1992 and 1997 and, more recently, the fastest growing areas are more concentrated in Central/Eastern Europe and Asia.

```
var func_Fastest_1pct = function(img){
  var img1 = img.select("scale");
  var RatePercentile = img1.reduceRegion( {
    reducer: ee.Reducer.percentile([95]),
    geometry: Regions.union().geometry(), // merge the regions into one feature to speed up computation
    scale: 5000, // to speed up the computing, I used larger scale in this analysis
    bestEffort: true,
    maxPixels: 1e9
  });
  var CutoffValue = ee.Number(RatePercentile.get("scale"));
  var Fastest_Growth = img1.mask(img1.gte(CutoffValue));
  Map.addLayer(Fastest_Growth, {palette:'#F012BE'}, "top5% fastest in the world");
};
```

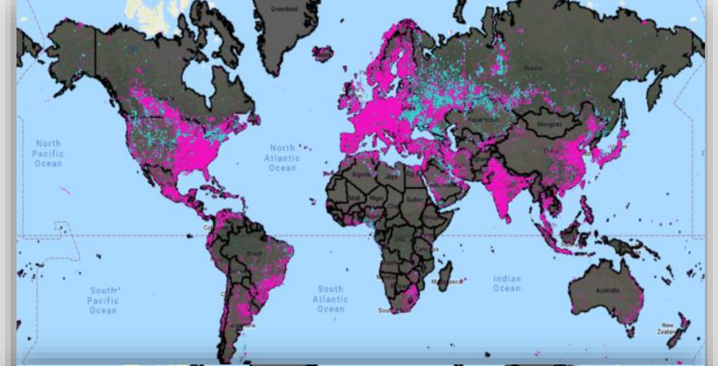
Growth or Decline



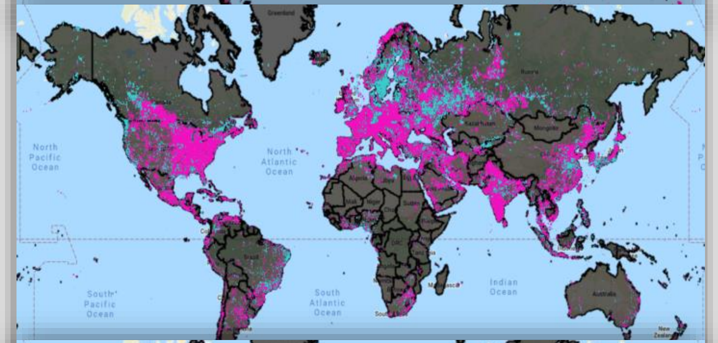
1992 – 2014

For further analysis, I divided the positive and negative change areas into image collections, red for positive, blue for negative. The dominance of blue between 2002 and 2007 is surprising. It may be caused by the Financial Crisis .

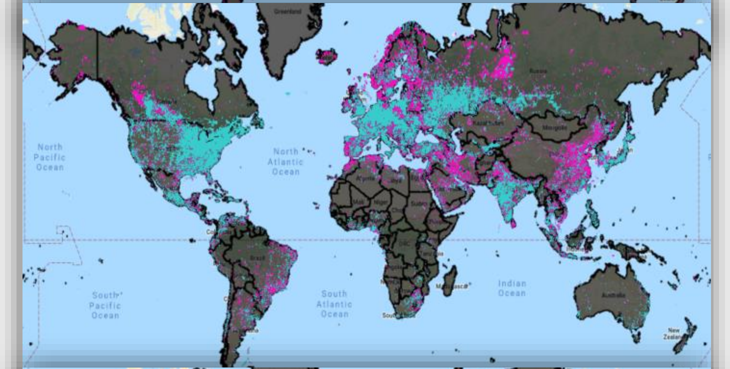
```
var func_PositiveGrowth = function(img){ // get the pixels with positive growth
  var img_scale = img.select("scale");
  var positive = img_scale.multiply(img_scale.gt(0));
  return positive;
};
var func_NegativeGrowth = function(img){ // get the pixels with positive growth
  var img_scale = img.select("scale");
  //transform the negative value into positive one for better visualization
  var negative = img_scale.multiply(img_scale.lt(0)).multiply(-1);
  return negative;
};
```



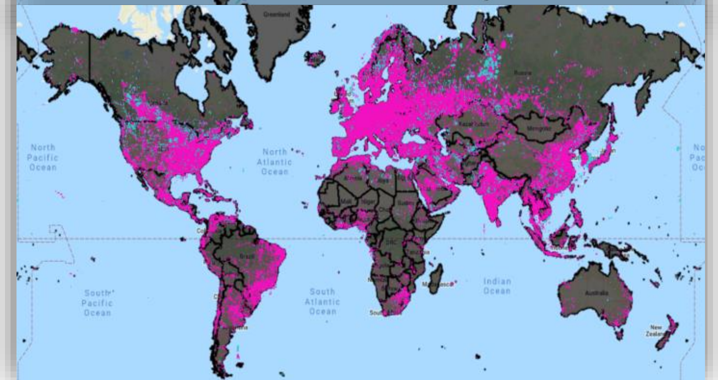
1992 – 1997



1997 – 2002

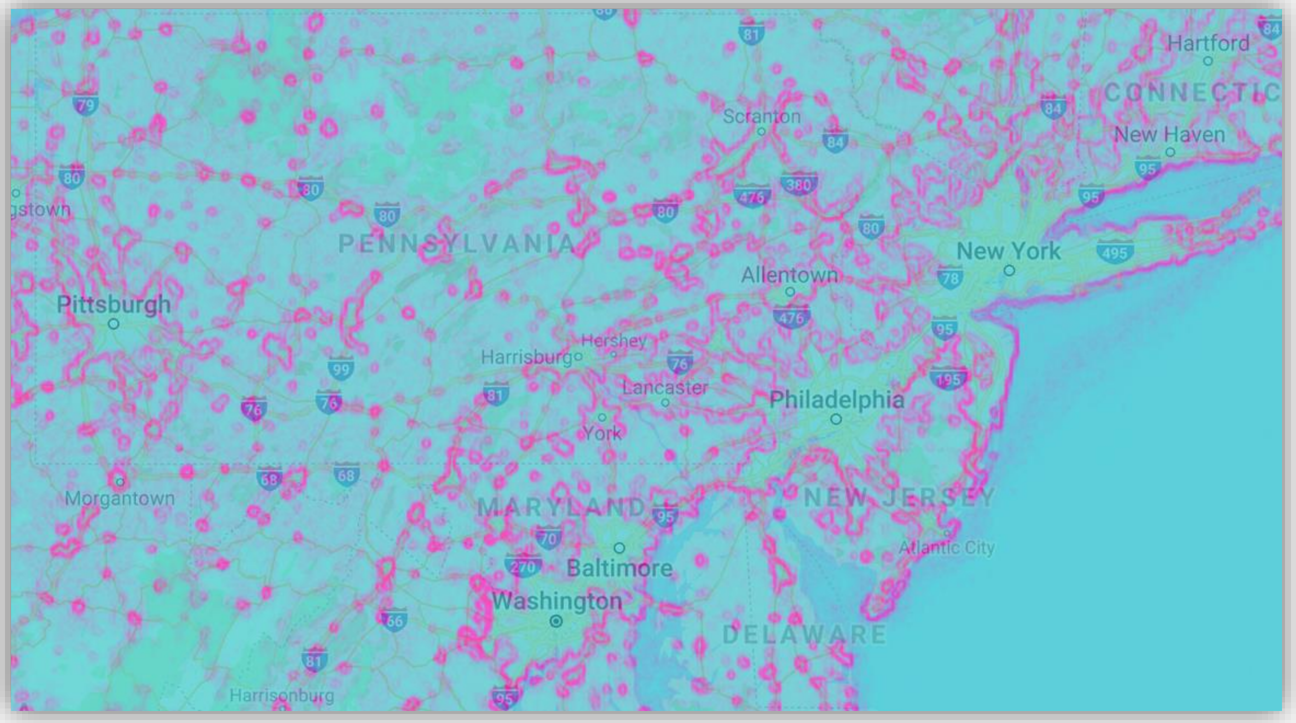


2002 – 2007



2007 – 2014

“Texture” of Brightness

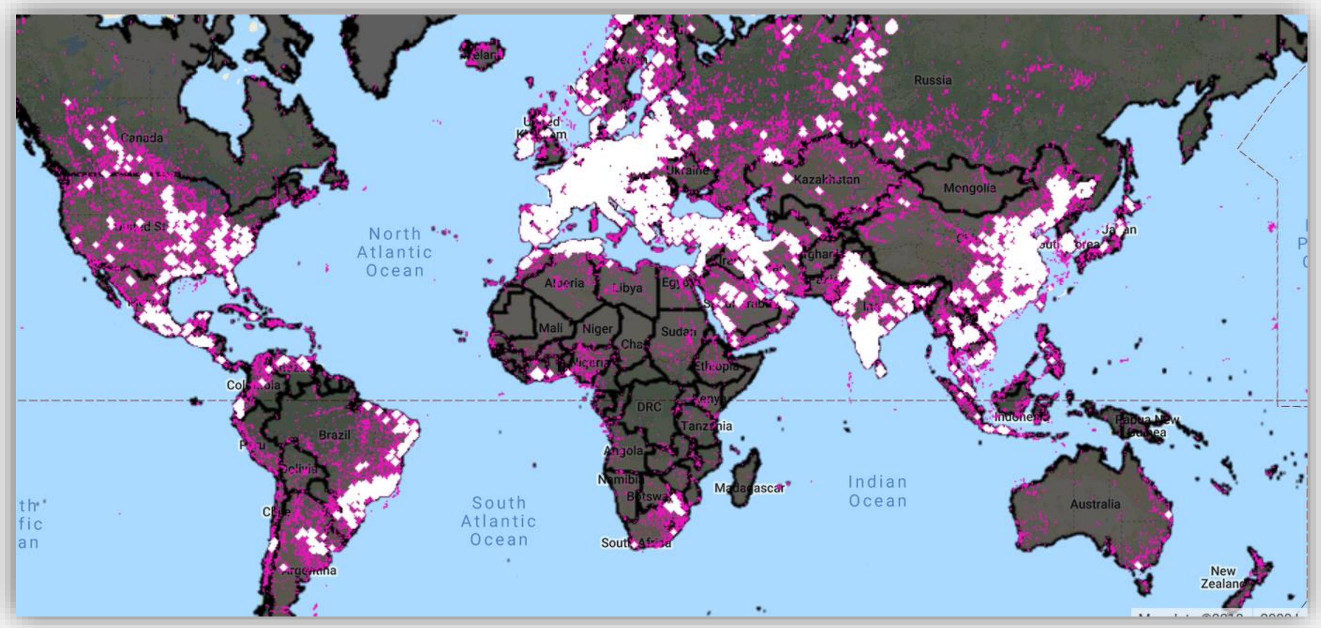


1992 – 2014

Scientists use standard deviation of NDVI in a neighborhood to represent the difference in texture between the logged area and protected area.
Here, I used SD of the intercept of the linear regression between 1992 and 2014 performed before in neighborhood to explore the “texture” of brightness. Big cities like NYC and Philadelphia are enclosed by red line, and the suburbs are more fractured.

```
//Texture of Brightness - variation
var fit_92_14_texture = fit_92_14.select("offset")
  .reduceNeighborhood({
    reducer: ee.Reducer.stdDev(),
    kernel: ee.Kernel.circle(1),
});
Map.addLayer(fit_92_14_texture,{min: 0, max: 10,
palette: ['#39CCCC', '#F012BE'], opacity: 0.7}, 'SD of Growth');
```

Concentrated Growth



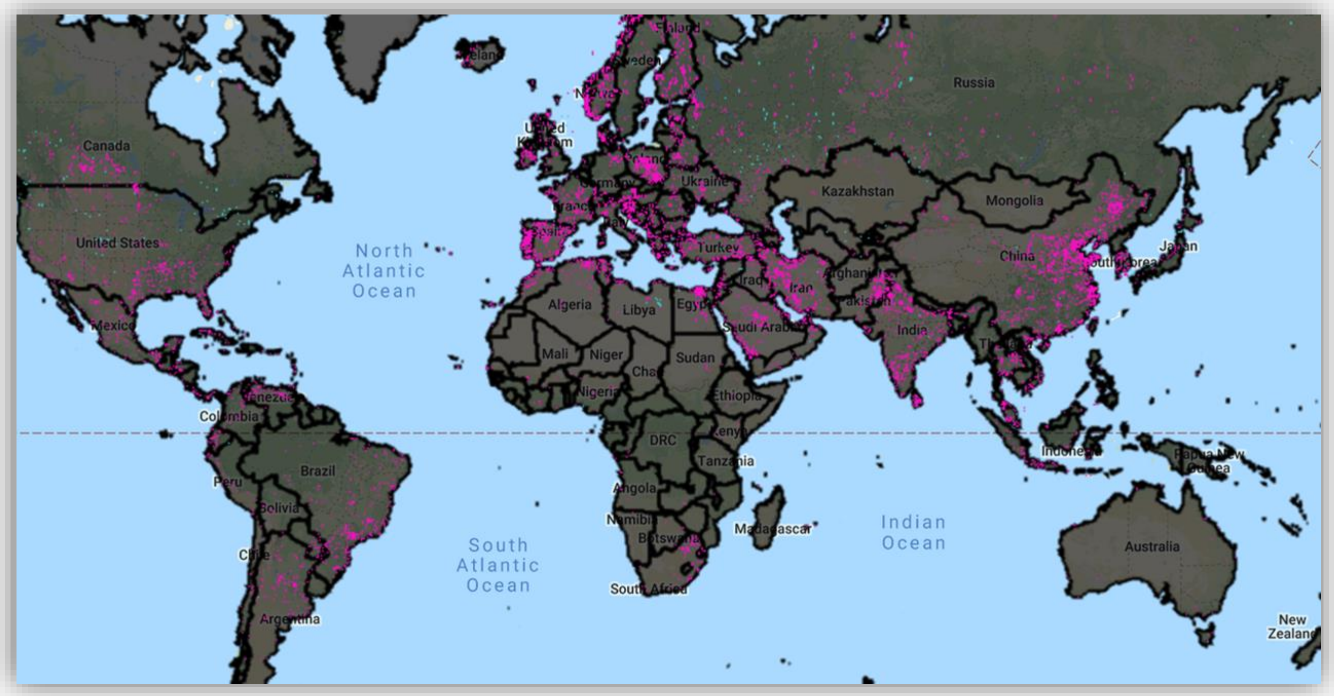
1992 – 2014

Human activities are agglomerated together, then is there any clustering pattern in areas experience growth in night-time light? From the result of opening operation, growth are concentrated in Europe, South/East Asia, and the east coast of the American Continent.

Opening operation - erosion and then dilation

```
//Identify the areas with agglomerated growth using opening
var kernel = ee.Kernel.circle({radius: 1});
var growth_92_14_p_opened = growth_92_14_p
                    .focal_min({kernel: kernel, iterations: 2})
                    .focal_max({kernel: kernel, iterations: 2});
```

Constant Growth Area

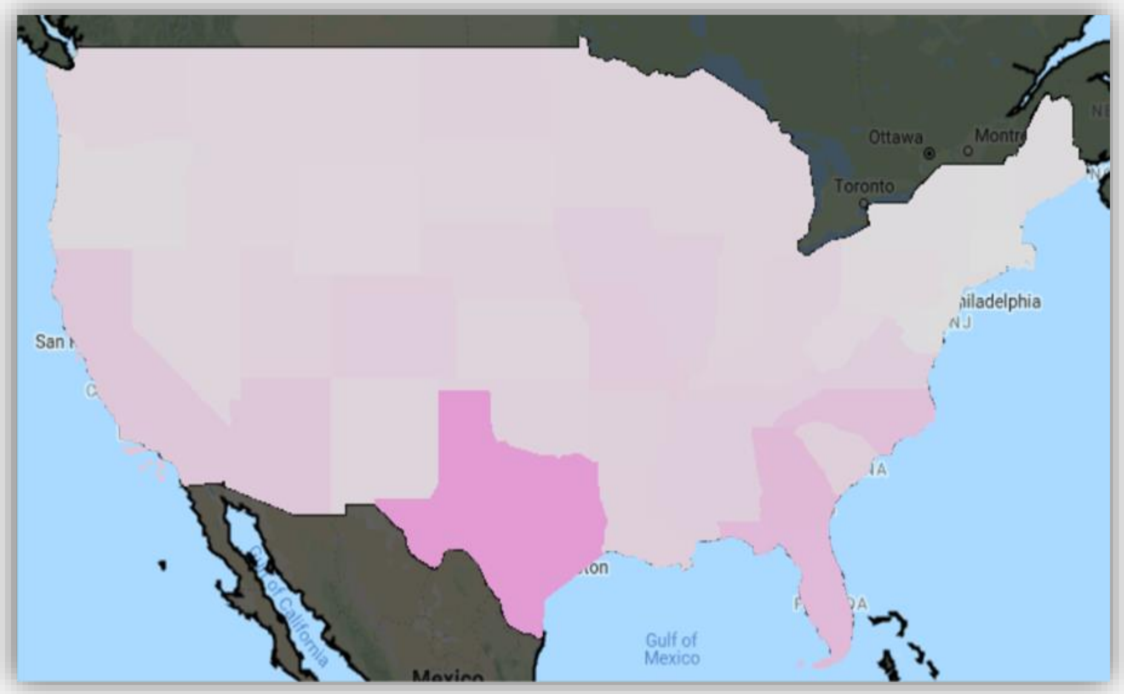


1992 – 1997, 1997 – 2002, 2002 – 2007, 2007 - 2014

I used *relational operation* to identify areas that have positive change rate in all four sub-periods as *constant growth area,* which are in red in the map. Area that have negative change rate in all four sub-periods are in blue.

```
//Identify the places experiening positive/negative growth in all four intervals
var const_growth_p = growth_92_97_p.and(growth_97_02_p)
                        .and(growth_02_07_p).and(growth_07_14_p);
Map.addLayer(const_growth_p.updateMask(const_growth_p), {palette: '#F012BE'});
var const_growth_n = growth_92_97_n.and(growth_97_02_n)
                        .and(growth_02_07_n).and(growth_07_14_n);
Map.addLayer(const_growth_n.updateMask(const_growth_n), {palette: '#39CCCC'});
```

Growth By States



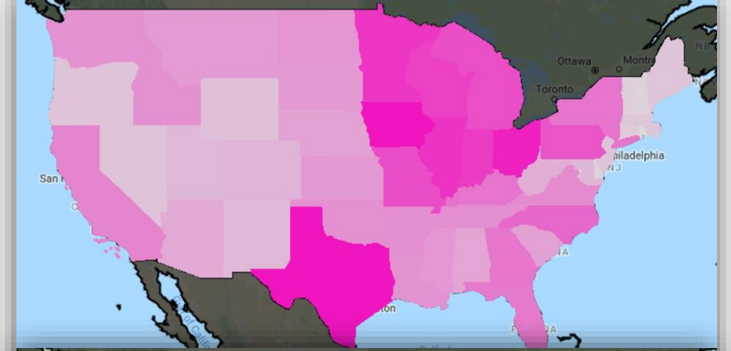
1992 – 2014

I calculated the sum of growth rate for each pixel inside each state in the U.S for different time intervals. In visualization, I used the same maximum value (15 billion) for all time intervals to make comparison across time easier.

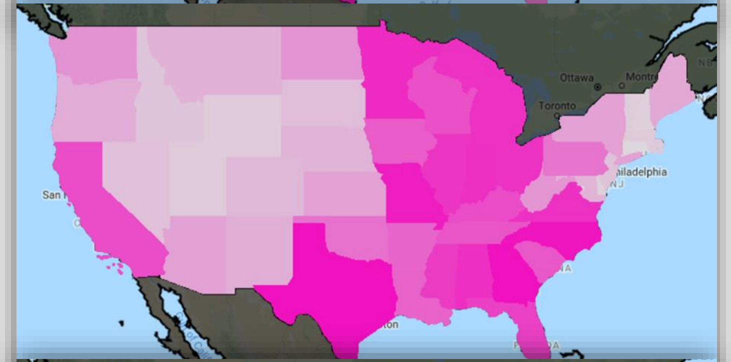
```
//Calculate and map the area with positive growth in each country
var area_92_14_p = states.map(function (f){
  var area = growth_92_14_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
```

Visualization Function

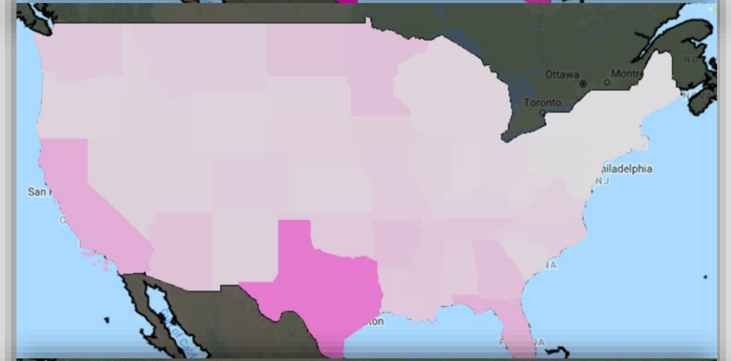
```
var func_MappingGrowth = function(f){
  var img_f = f.reduceToImage( ['LitArea'], ee.Reducer.first());
  Map.addLayer(img_f,
    {max: 15000000000000, palette:['DDDDDD','39CCCC'], opacity:1},
    'Positive Growth Area Lit' );
};
func_MappingGrowth(area_92_14_p);
```



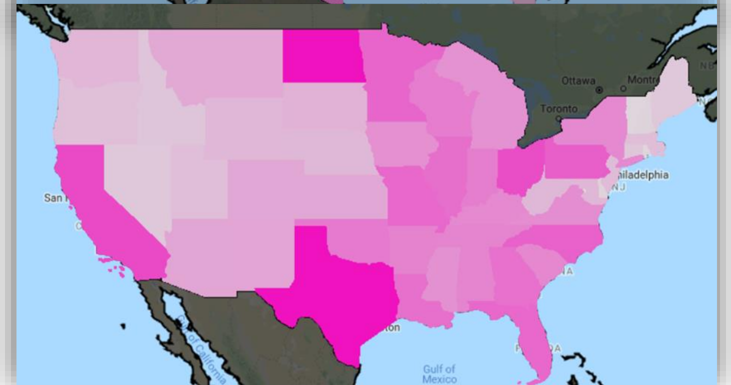
1992 – 1997



1997 – 2002

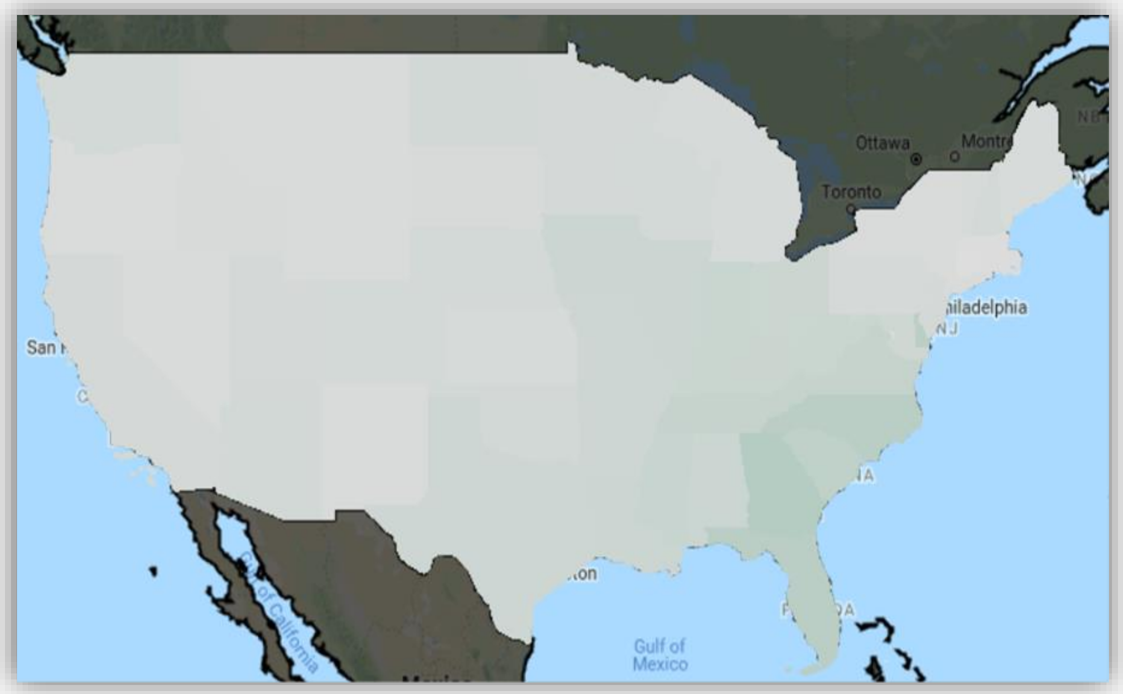


2002 – 2007



2007 – 2014

Growth Density Across States



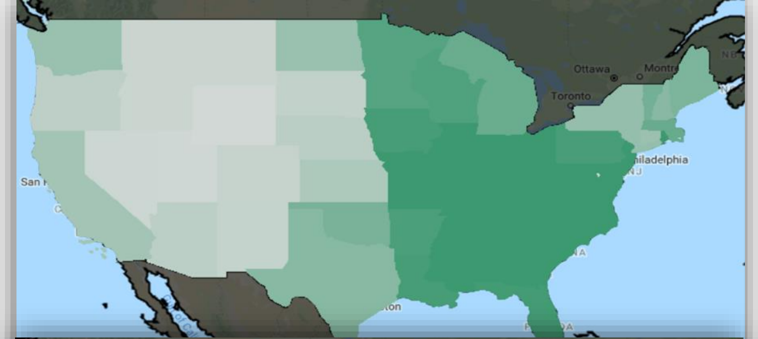
1992 – 2014

I calculated growth density by normalizing the sum of growth rate in every states by its area to adjust for the influence of area.

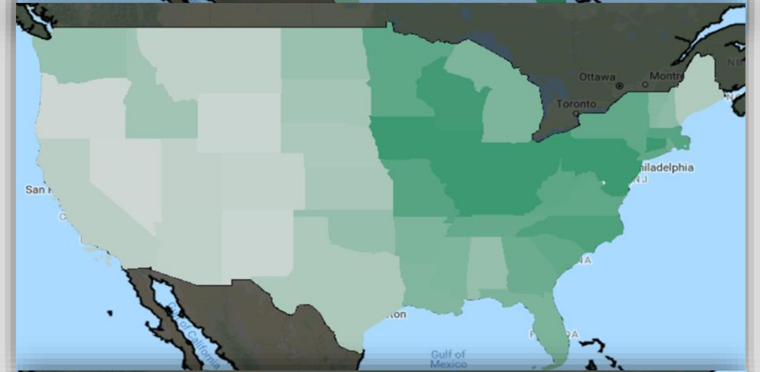
```
var func_Density = function(f){
  var Area = ee.Number(f.area());
  var LitDensity = ee.Number(f.get('LitArea')).divide(Area);
  return f.set({'LitDensity': LitDensity});
};
```

Visualization Function

```
var func_MappingGrowthDensity = function(f){
  var img_f = f.reduceToImage( ['LitDensity'], ee.Reducer.first());
  Map.addLayer(img_f, {max: 70, palette:['DDDDDD','39CCCC'], opacity:1}, 'Positive Growth Density' );
};
```

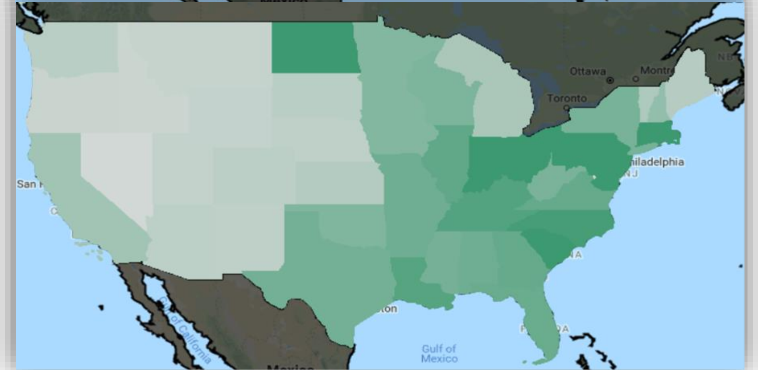


1992 – 1997



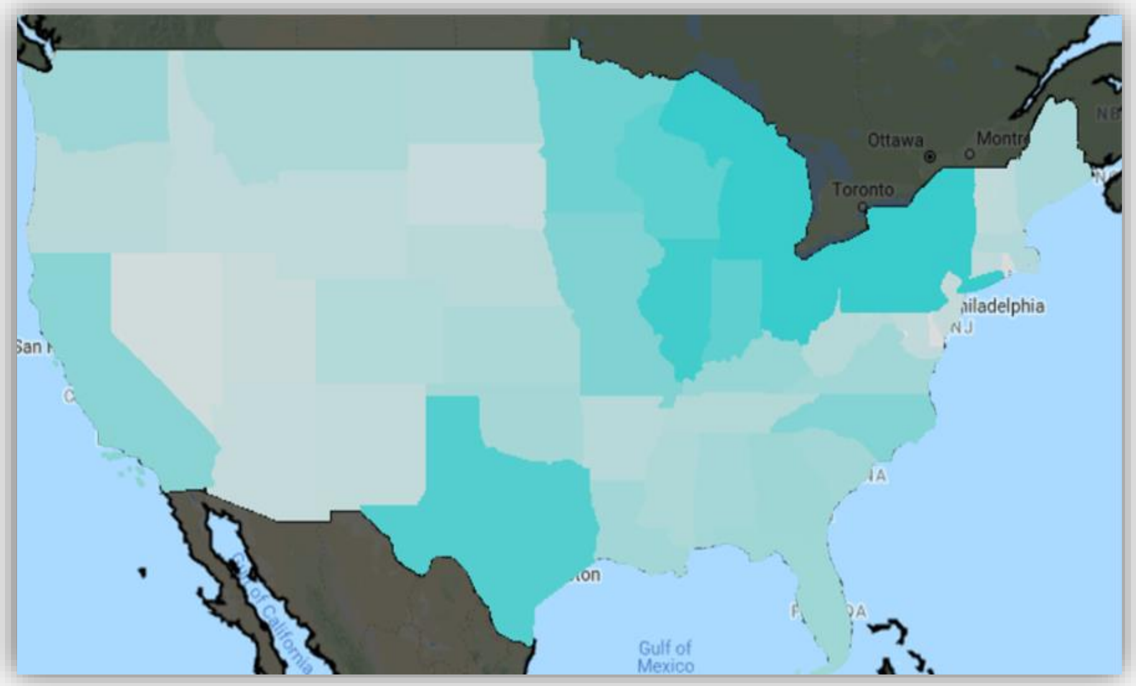
1997 – 2002

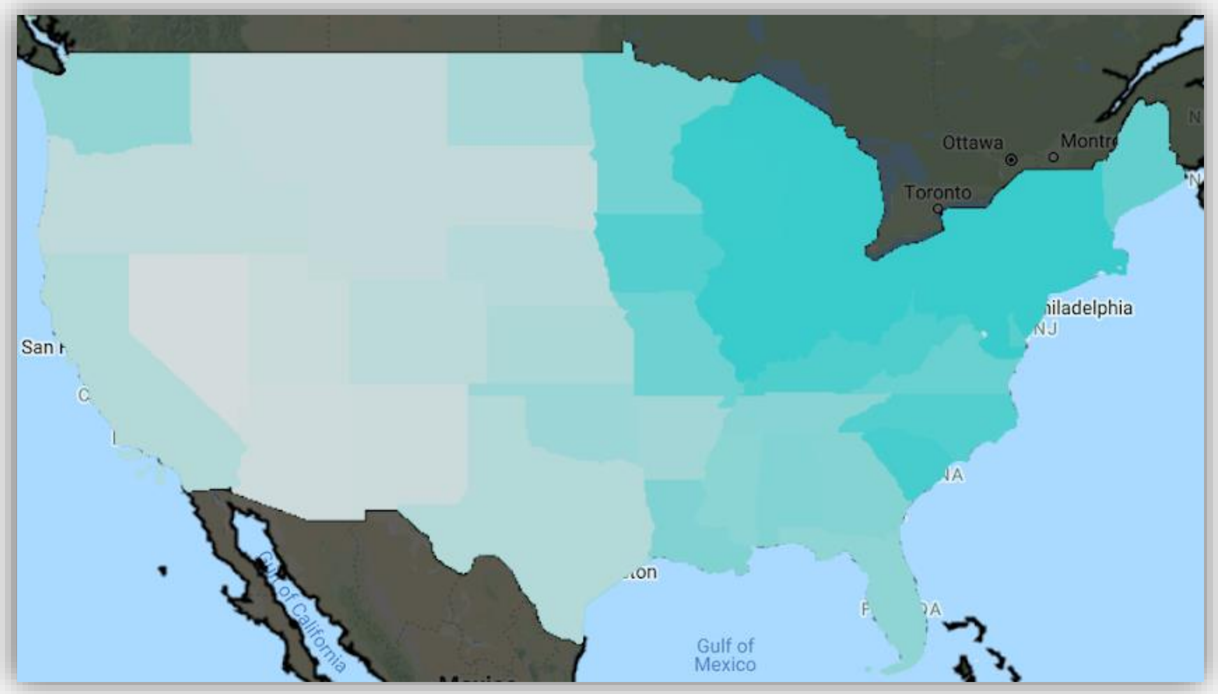
2002 – 2007



2007 – 2014

Dawn of Financial Crisis: 2002 - 2007





The two maps above are the *decline rate map* and *decline density map* calculating by same method as growth density for 2002 and 2007. The Northeast seems to be the place experiencing the largest decline of human activities.

```
var area_02_07_n = states.map(function (f){
  var area = growth_02_07_n.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
```

Full Script

```
//1. Data Preparation
//1.1 Load vector dataset regions in the world and states in the U.S. as FeatureCollections

var Regions = ee.FeatureCollection("ft:1PRMJUI4OourlypJaFOHd351LfgEfsvrMJ2_9sPQV"); // Vector file of
Countries
var states = ee.FeatureCollection('ft:17aT9Ud-YnGiXdXEJUyycH2ocUqreOeKGbzCkUw'); // Vector file of states in
the US
print(states.getInfo());

//1.2 Load Night-time lights data as ImageCollection
//Reference to "Night-time lights: A global, long term look at links to socio-economic trends" by Proville
et al., 2017
//Credit to Jeremy Proville, Environmental Defense Fund

var blank = ee.Image(0);
//Select the "stable_lights" band - The cleaned up average of the visible band digital number values
//contains the lights from cities, towns, and other sites with persistent lighting.
//The background noise was identified and replaced with values of zero.
var World_all = ee.ImageCollection('NOAA/DMSP-OLS/NIGHTTIME_LIGHTS').select("stable_lights");
Map.addLayer(World_all, {min: 0, max: 60, palette:['#111111','#39CCCC']}, 'Night-time light in the World');
```

Also could access through:
https://code.earthengine.google.com/323a9d039d088de2a8b8a02efd58553c

```
//Find the place where are considered "lit" - a digital number greater than 31
//The values in the image range from 0 to 63
function AreaLit(i){
  var thold = blank.where(i.gt(31),1); //Set the threshold of lit - 31
  var alit = thold.mask(thold);
  alit = ee.Image.pixelArea().mask(alit);
  alit = alit.divide(1000000);
  alit = alit.set('index',i.get('system:index'));
  return alit;
  }
function tabulate(i){
  return Regions.map(function (f){
    var r = i.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort:true,
      maxPixels:1e9});
    return ee.Feature(null, {
      name: f.get('name'),
      area: r.get('area'),
      index: i.get('index')});
  });}

var AreaLit_Results = World_all.map(AreaLit); //A ImageCollection containing the lit area
var AreaLit_ResultsTable = AreaLit_Results.map(tabulate).flatten(); //A table with lit area info in every countries
Export.table.toDrive(AreaLit_ResultsTable,'DMSP_results','DMSP results','results','csv');
//Export the table to Google drive in .csv. Full dataset took over 2 hours
//If just focus on one region, consider using "ee.Filter.eq('Name', 'NameofRegion')" to the FeatureCollection
//If just focus on a time interval, consider using ".filterDate()" to narrow down time
```

```
//Map regions in the world and lit area
Map.addLayer(Regions, {opacity: 0.6}, 'World Regions');
Map.addLayer(AreaLit_Results.select("area"), {palette:'#39CCCC'}, 'Lit Area in the World, DN >31');

// Add a band containing image date as years since 1990.
var createTimeBand_fromLinearRegression = function(image) {
  return image.addBands(image.metadata('system:time_start').divide(1e18));
};

var GetYear = function(y){
  return ee.Date.fromYMD({
    day: 1, month: 1, year: y});
};
var func_LinearFit = function(year1, year2){
  var SYear = GetYear(year1);
  var EYear = GetYear(year2);
  var world_TimeBand = World_all.filterDate(SYear, EYear).map(createTimeBand_fromLinearRegression);
  var linearFit = world_TimeBand.select(['system:time_start','stable_lights'])
    .reduce(ee.Reducer.linearFit());
  return linearFit;
};

var fit_92_14 = func_LinearFit(1992, 2014);
var fit_92_97 = func_LinearFit(1992, 1997);
var fit_97_02 = func_LinearFit(1997, 2002);
var fit_02_07 = func_LinearFit(2002, 2007);
var fit_07_14 = func_LinearFit(2007, 2014);
```

```
Map.addLayer(fit_92_14, {min:0, max: [0.18, 20, -0.18], bands: ['scale','offset', 'scale']}, '1992-2010’);
Map.addLayer(fit_92_97, {min:0, max: [0.18, 20, -0.18], bands: ['scale','offset', 'scale']}, '1992-1997');
Map.addLayer(fit_97_02, {min:0, max: [0.18, 20, -0.18], bands: ['scale','offset', 'scale']}, '1997-2002');
Map.addLayer(fit_02_07, {min:0, max: [0.18, 20, -0.18], bands: ['scale','offset', 'scale']}, '2002-2007');
Map.addLayer(fit_07_14, {min:0, max: [0.18, 20, -0.18], bands: ['scale','offset', 'scale']}, '2007-2014');


//Identify the fastest growing 5% area
var func_Fastest_1pct = function(img){
  var img1 = img.select("scale");
  var RatePercentile = img1.reduceRegion( {
    reducer: ee.Reducer.percentile([95]),
    geometry: Regions.union().geometry(), // merge the regions into one feature to speed up computation
    scale: 5000, // to speed up the computing, I used larger scale in this analysis
    bestEffort: true,
    maxPixels: 1e9
  });
  var CutoffValue = ee.Number(RatePercentile.get("scale"));
  var Fastest_Growth = img1.mask(img1.gte(CutoffValue));
  Map.addLayer(Fastest_Growth, {palette:'#F012BE'}, "top5% fastest in the world");
};
func_Fastest_1pct(fit_92_14);
func_Fastest_1pct(fit_92_97);
func_Fastest_1pct(fit_07_14);
```

```
var func_PositiveGrowth = function(img){ // get the pixels with positive growth
  var img_scale = img.select("scale");
  var positive = img_scale.multiply(img_scale.gt(0));
  return positive;
};
var func_NegativeGrowth = function(img){ // get the pixels with negative growth
  var img_scale = img.select("scale");
  //transform the negative value into positive one for better visualization
  var negative = img_scale.multiply(img_scale.lt(0)).multiply(-1);
  return negative;
};
var growth_92_14_p = func_PositiveGrowth(fit_92_14);
var growth_92_14_n = func_NegativeGrowth(fit_92_14);
var growth_92_97_p = func_PositiveGrowth(fit_92_97);
var growth_92_97_n = func_NegativeGrowth(fit_92_97);
var growth_97_02_p = func_PositiveGrowth(fit_97_02);
var growth_97_02_n = func_NegativeGrowth(fit_97_02);
var growth_02_07_p = func_PositiveGrowth(fit_02_07);
var growth_02_07_n = func_NegativeGrowth(fit_02_07);
var growth_07_14_p = func_PositiveGrowth(fit_07_14);
var growth_07_14_n = func_NegativeGrowth(fit_07_14);

var func_MappingGrowth_Global = function(imgP, imgN){
  Map.addLayer(imgP.updateMask(imgP), {min:0, max: 0.18, palette: '#F012BE'}, 'positive');
  Map.addLayer(imgN.updateMask(imgN), {min:0, max: 0.18, palette: '#39CCCC'}, 'negative');
};
```

```
func_MappingGrowth_Global(growth_92_14_p,growth_92_14_n);
func_MappingGrowth_Global(growth_92_97_p,growth_92_97_n);
func_MappingGrowth_Global(growth_97_02_p,growth_97_02_n);
func_MappingGrowth_Global(growth_02_07_p,growth_02_07_n);
func_MappingGrowth_Global(growth_07_14_p,growth_07_14_n);

//Identify the areas with agglomerated growth using opening - erosion and dilation
var kernel = ee.Kernel.circle({radius: 1}); // Define a kernel.
var growth_92_14_p_opened = growth_92_14_p // Perform an erosion followed by a dilation
             .focal_min({kernel: kernel, iterations: 2})
             .focal_max({kernel: kernel, iterations: 2});
Map.addLayer(growth_92_14_p_opened.updateMask(growth_92_14_p_opened), {}, 'opened');

//Texture of Brightness - variation
var fit_92_14_texture = fit_92_14.select("offset")
  .reduceNeighborhood({
    reducer: ee.Reducer.stdDev(),
    kernel: ee.Kernel.circle(1),
});
Map.addLayer(fit_92_14_texture,{min: 0, max: 10, palette: ['#39CCCC', '#F012BE'], opacity: 0.7}, 'SD of Growth');


//Identify the places experiening positive/negative growth in all four intervals
var const_growth_p = growth_92_97_p.and(growth_97_02_p)
                       .and(growth_02_07_p).and(growth_07_14_p);
Map.addLayer(const_growth_p.updateMask(const_growth_p), {palette: '#F012BE'});
var const_growth_n = growth_92_97_n.and(growth_97_02_n)
                       .and(growth_02_07_n).and(growth_07_14_n);
Map.addLayer(const_growth_n.updateMask(const_growth_n), {palette: '#39CCCC'});
```

```
//Focusing in the US
//Calculate and map the area with positive growth in each country
var area_92_14_p = states.map(function (f){
  var area = growth_92_14_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
var area_92_97_p = states.map(function (f){
  var area = growth_92_97_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
var area_97_02_p = states.map(function (f){
  var area = growth_97_02_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
```

```
var area_02_07_p = states.map(function (f){
  var area = growth_02_07_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
var area_07_14_p = states.map(function (f){
  var area = growth_07_14_p.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
var area_02_07_n = states.map(function (f){
  var area = growth_02_07_n.reduceRegion({
      reducer: ee.Reducer.sum(),
      geometry: f.geometry(),
      scale: 500,
      bestEffort: true,
      maxPixels: 1e9});
  var f_copy = f;
  return f_copy.set({LitArea: area.get('scale')});
});
```

```
//Technical Explanation for these huge functions:
//I noticed that creating functions for every time interval respectively
//can drastically reduce computing time comparing to creating one two-parameter function

var func_MappingGrowth = function(f){
  var img_f = f.reduceToImage( ['LitArea'], ee.Reducer.first());
  Map.addLayer(img_f, {max: 15000000000000, palette:['DDDDDD','39CCCC'], opacity:1}, 'Positive Growth Area Lit' );
};
func_MappingGrowth(area_92_14_p);
func_MappingGrowth(area_92_97_p);
func_MappingGrowth(area_97_02_p);
func_MappingGrowth(area_02_07_p);
func_MappingGrowth(area_07_14_p);
func_MappingGrowth(area_92_10_n);
func_MappingGrowth(area_92_97_n);
func_MappingGrowth(area_97_02_n);
func_MappingGrowth(area_02_07_n);
func_MappingGrowth(area_07_14_n);

//Calculate and map the Density = (area with positive/negative growth/total area)
var func_Density = function(f){
  var Area = ee.Number(f.area());
  var LitDensity = ee.Number(f.get('LitArea')).divide(Area);
  return f.set({'LitDensity': LitDensity});
};
var density_92_14_p = area_92_14_p.map(func_Density);
var density_92_97_p = area_92_97_p.map(func_Density);
var density_97_02_p = area_97_02_p.map(func_Density);
var density_02_07_p = area_02_07_p.map(func_Density);
var density_07_14_p = area_07_14_p.map(func_Density);
```

```
var density_92_14_p = area_92_14_p.map(func_Density);
var density_92_97_p = area_92_97_p.map(func_Density);
var density_97_02_p = area_97_02_p.map(func_Density);
var density_02_07_p = area_02_07_p.map(func_Density);
var density_07_14_p = area_07_14_p.map(func_Density);
var density_92_10_n = area_92_10_n.map(func_Density);
var density_92_97_n = area_92_97_n.map(func_Density);
var density_97_02_n = area_97_02_n.map(func_Density);
var density_02_07_n = area_02_07_n.map(func_Density);
var density_07_14_n = area_07_14_n.map(func_Density);

var func_MappingGrowthDensity = function(f){
  var img_f = f.reduceToImage( ['LitDensity'], ee.Reducer.first());
  Map.addLayer(img_f, {max: 70, palette:['DDDDDD','39CCCC'], opacity:1}, 'Positive Growth Density' );
};
func_MappingGrowthDensity(density_92_14_p);
func_MappingGrowthDensity(density_92_97_p);
func_MappingGrowthDensity(density_97_02_p);
func_MappingGrowthDensity(density_02_07_p);
func_MappingGrowthDensity(density_07_14_p);
func_MappingGrowthDensity(density_92_10_n);
func_MappingGrowthDensity(density_92_97_n);
func_MappingGrowthDensity(density_97_02_n);
func_MappingGrowthDensity(density_02_07_n);
func_MappingGrowthDensity(density_07_14_n);
```

Reference

Proville, J., Zavala-Araiza, D. and Wagner, G., 2017. Night-time lights: A global, long term look at links to socio-economic trends. *PloS one*, *12*(3), p.e0174610.

Sun, Yuting. 2016. CPLN670 Final Project

Qu, Ge. 2016. CPLN670 Final Project

Google Earth Engine API: https://developers.google.com/earth-engine

Special Thanks to Evan for everything