GIS FOR ECONOMICS RESEARCH

Masayuki Kudamatsu

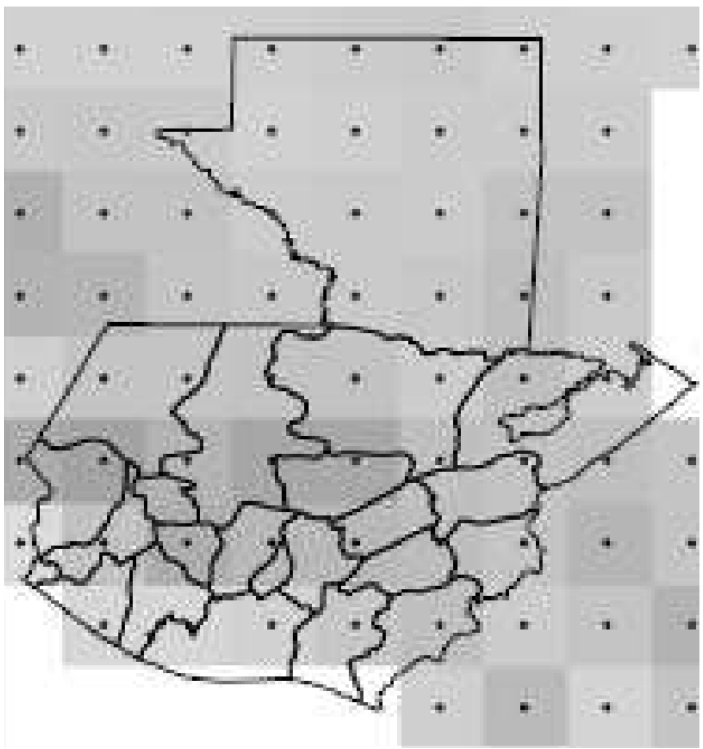
14-15 June, 2016

WHY GIS FOR ECONOMICS?

Reason 1: GIS makes more research feasible

GIS MAKES RESEARCH FEASIBLE (1/3)

BY MERGING DATA BY LOCATION



(Figure 2.3 of Dell 2009)

GIS MAKES RESEARCH FEASIBLE (2/3)

BY SCANNED OLD MAPS

e.g. Ethnic homelands in Africa by Murdock (1959)

Digitized by Nunn (2008)

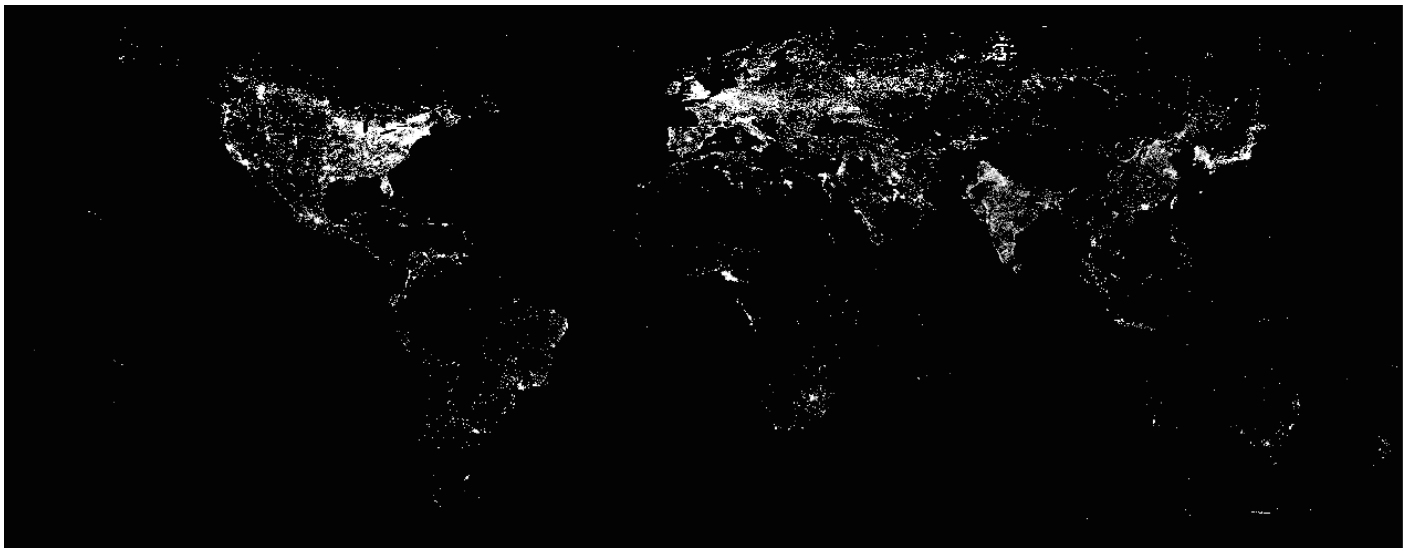
Used by Nunn & Wantchekon (2011), Michalopoulos & Papaioannou (2013, 2014, 2015), Alsan (2015), Alesina et al. (2016), etc.

(Figure 5A of Alsan (2015))

GIS MAKES RESEARCH FEASIBLE (3/3)

BY SATELLITE IMAGES

e.g. DMSP-OLS Nighttime Lights



Used by Henderson et al (2012), Pinkovskiy & Sala-i-Martin (2016), Hodler & Raschky (2014), Michalopoulos & Papaioannou (2013, 2014), Alesina et al (2016), etc.

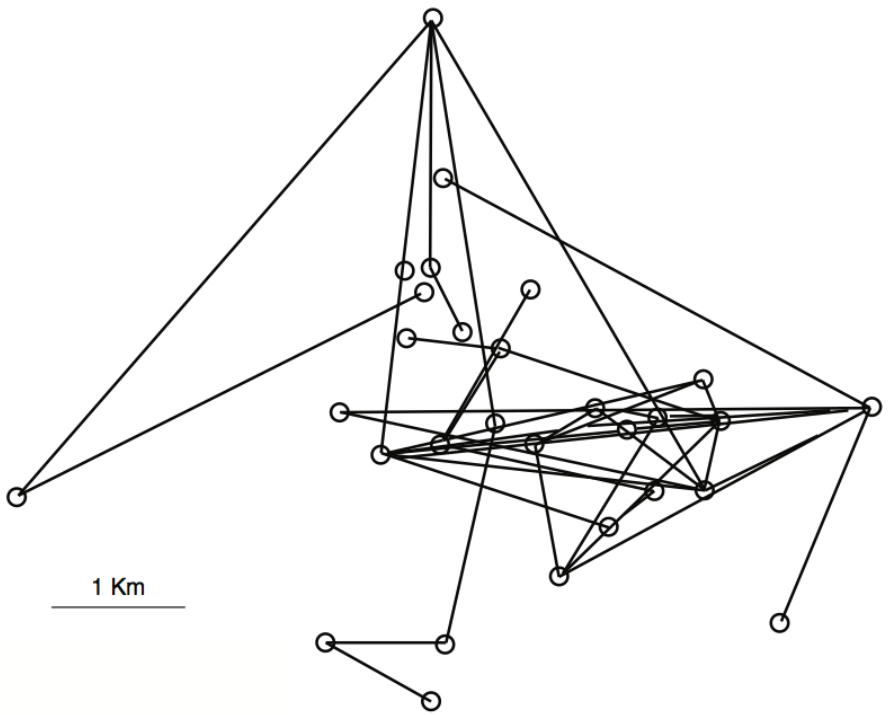
WHY GIS FOR ECONOMICS?

Reason 2: GIS makes identification more credible

GIS MAKES IDENTIFICATION CREDIBLE (1/4)

BY CONTROLLING FOR MORE COVARIATES

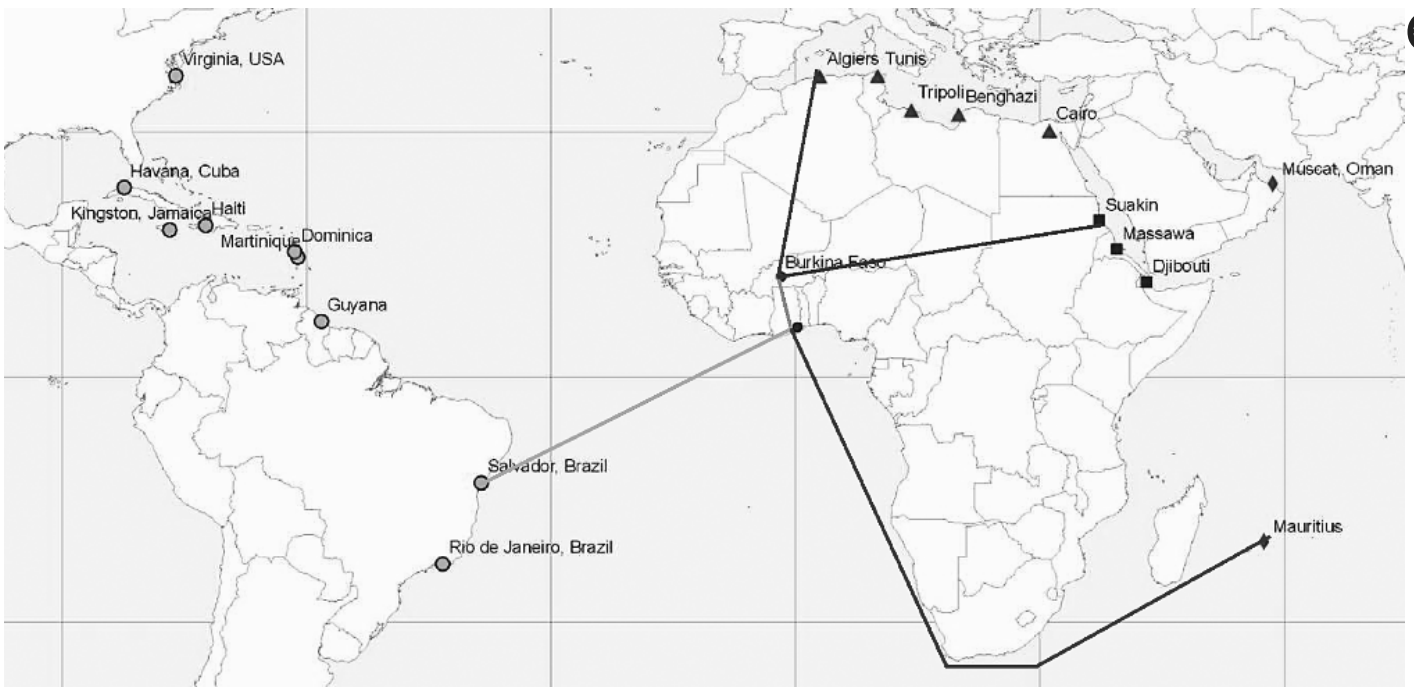
e.g. Peer effect estimation



(Figure 4 of Conley & Udry (2010))

GIS MAKES IDENTIFICATION CREDIBLE (2/4)

BY CONSTRUCTING INSTRUMENTS

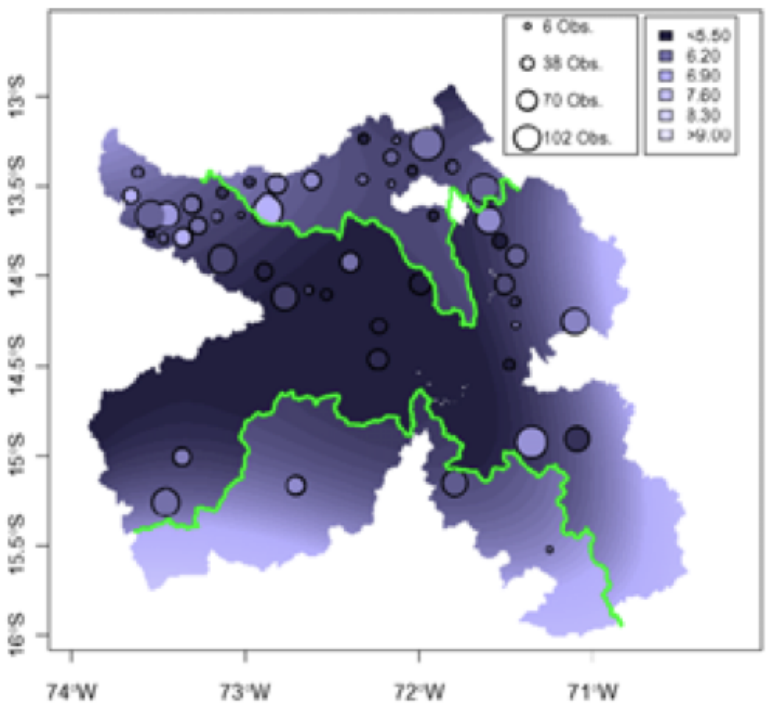


e.g. Distance

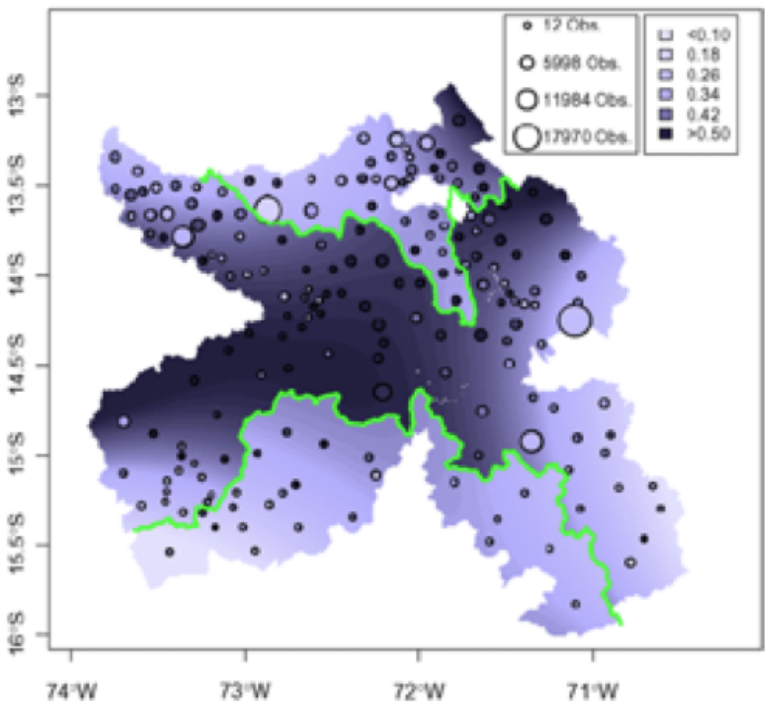
(Figure 5 of Nunn 2008)

GIS MAKES IDENTIFICATION CREDIBLE (3/4)

BY REGRESSION DISCONTINUITY



(a) Consumption (2001)

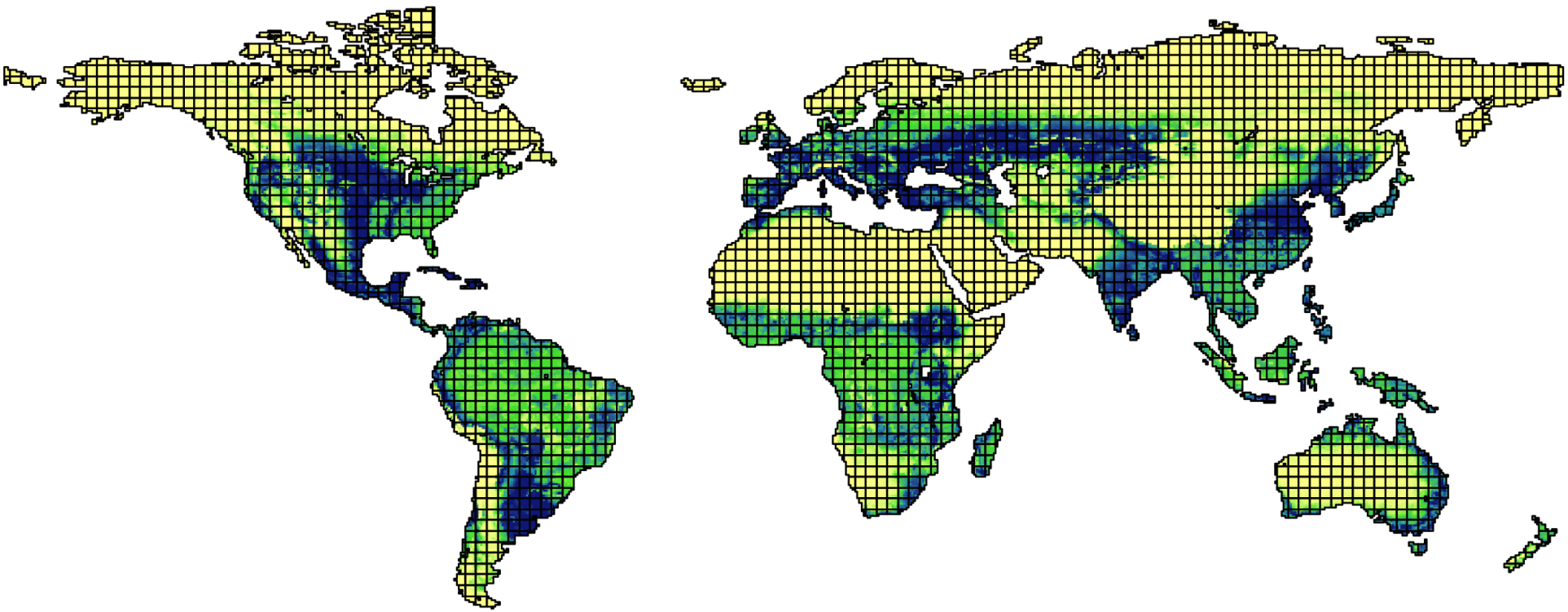


(b) Stunting (2005)

(Figure 2 of Dell (2010))

GIS MAKES IDENTIFICATION CREDIBLE (4/4)

BY EXOGENOUS BOUNDARIES



(image source)

ROAD MAP

1. GIS basics
2. Create spatial datasets on your own
3. Merge spatial datasets
4. Elevation
5. Distance
6. Spatial regression discontinuity
7. Surface area
8. Map Algebra

1. GIS BASICS

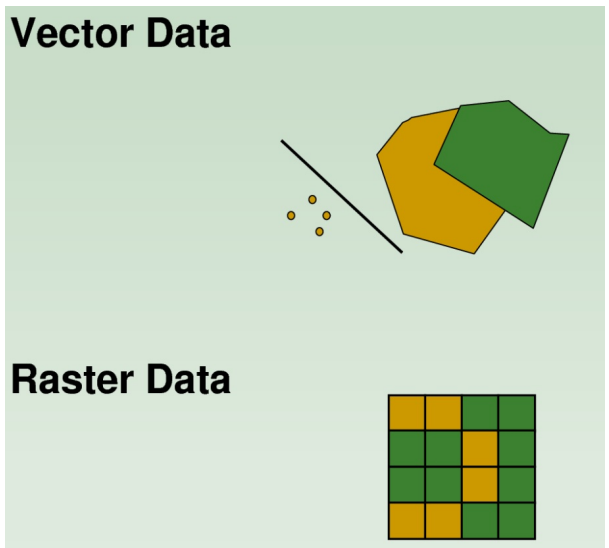
Data type

Coordinate systems

GIS software

1.1 DATA TYPE

Spatial data comes in two different formats: **Vector** & **Raster**



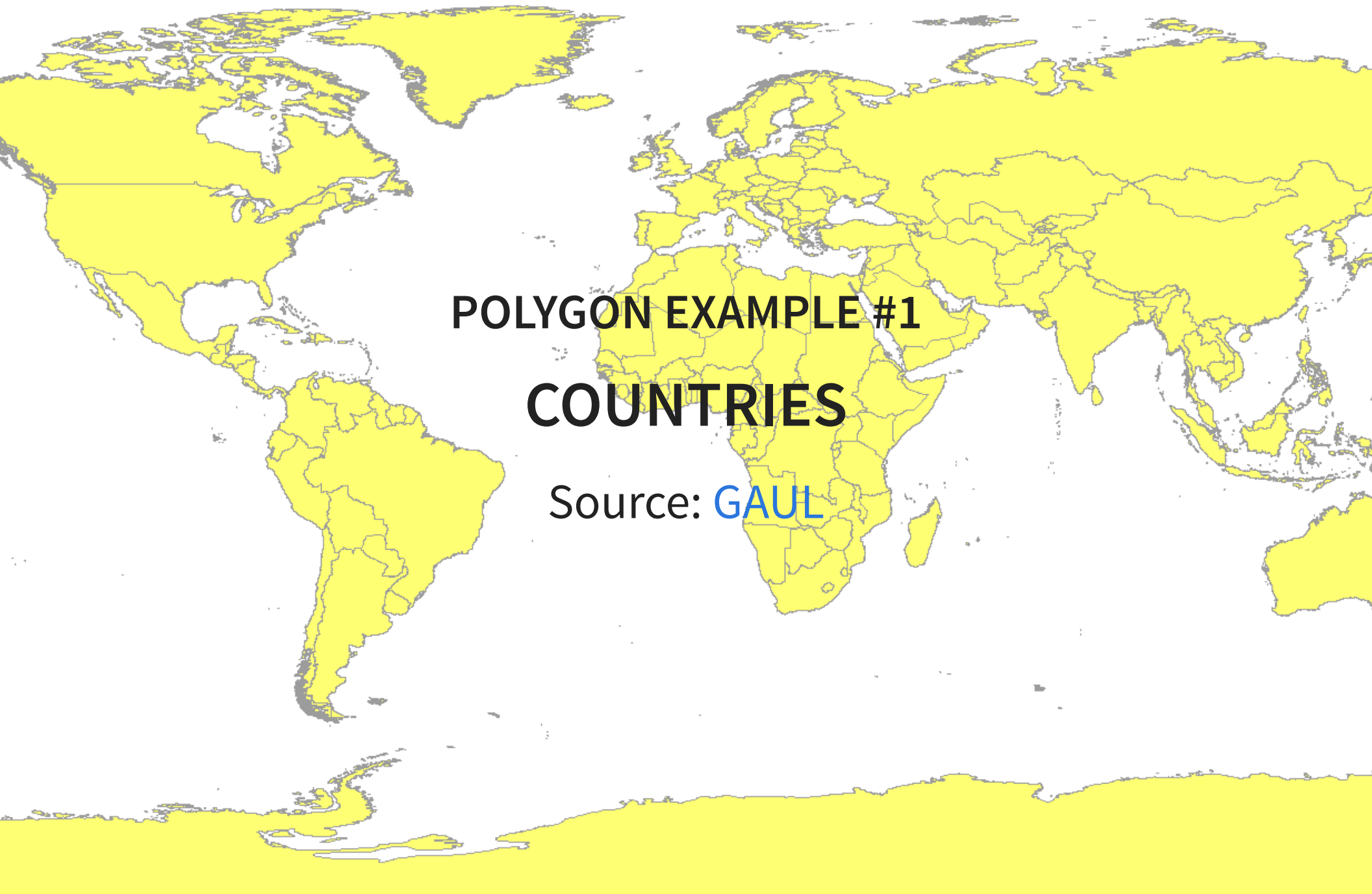
How to edit data differs a lot between them

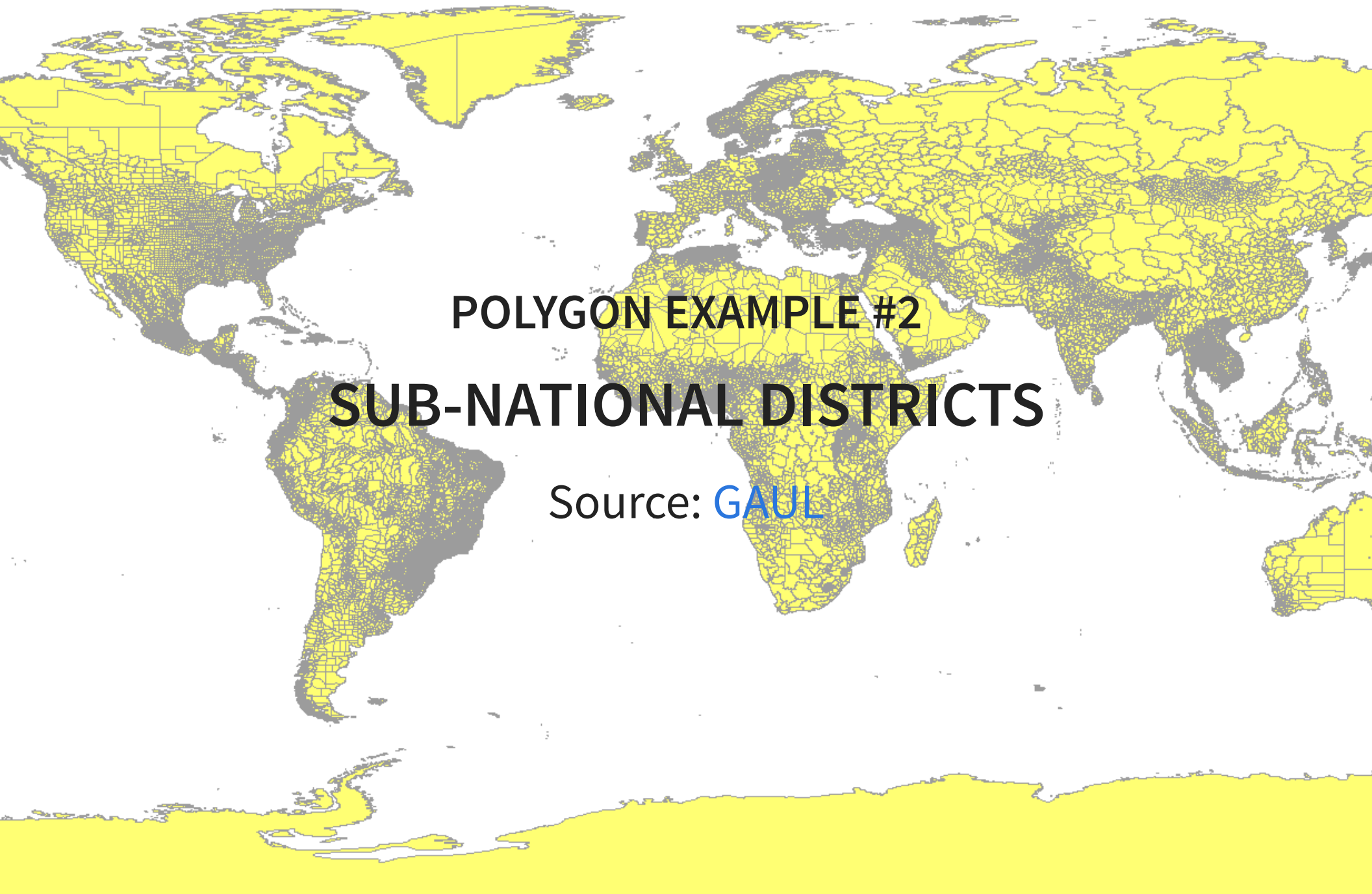
VECTOR DATA

Comes in three formats:

- Polygons
- Polylines
- Points

File format: **Shapefile** (.shp)

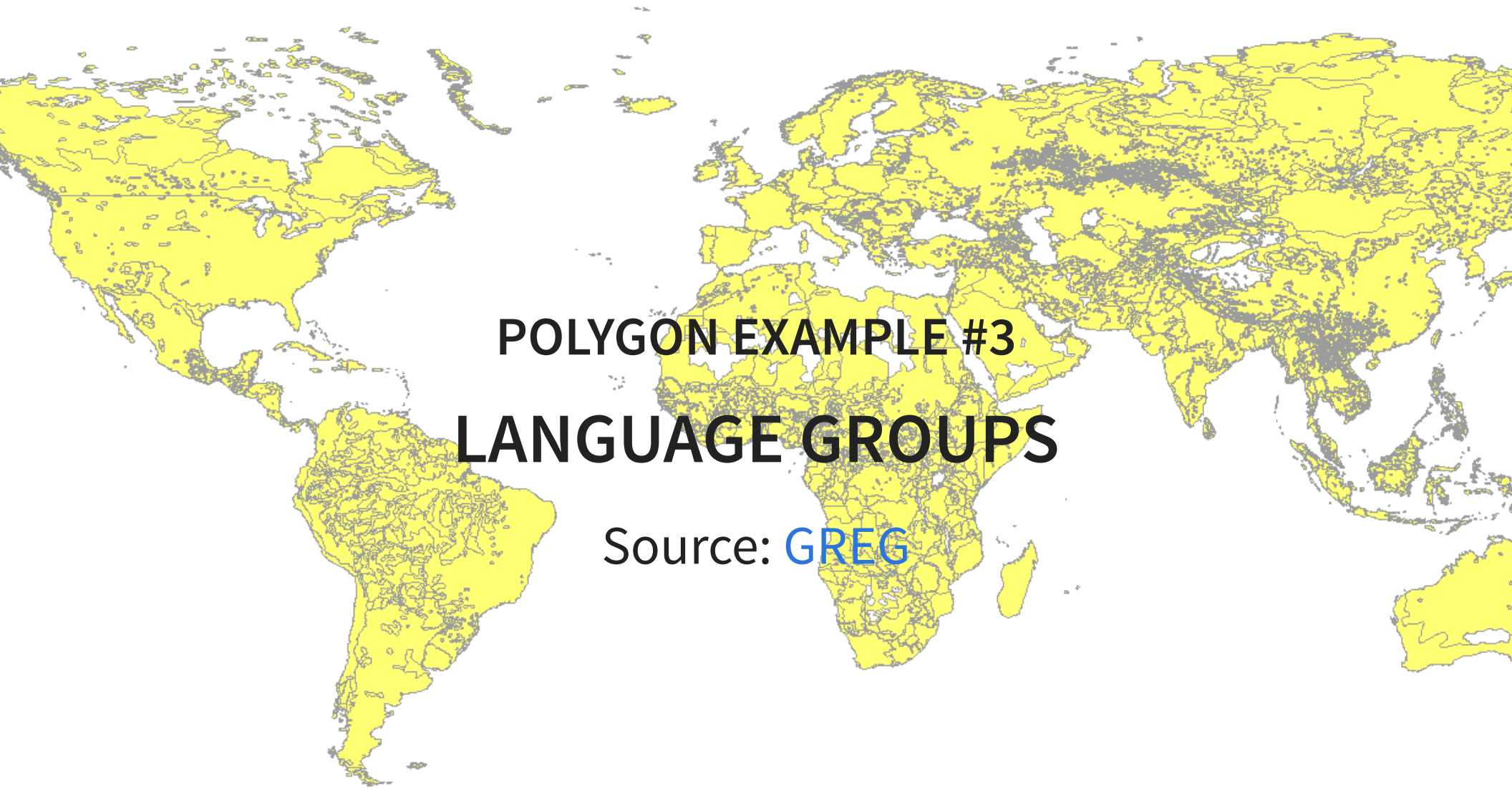

POLYGON EXAMPLE #1
COUNTRIES
Source: GAUL



POLYGON EXAMPLE #2

SUB-NATIONAL DISTRICTS

Source: GAUL



POLYGON EXAMPLE #3

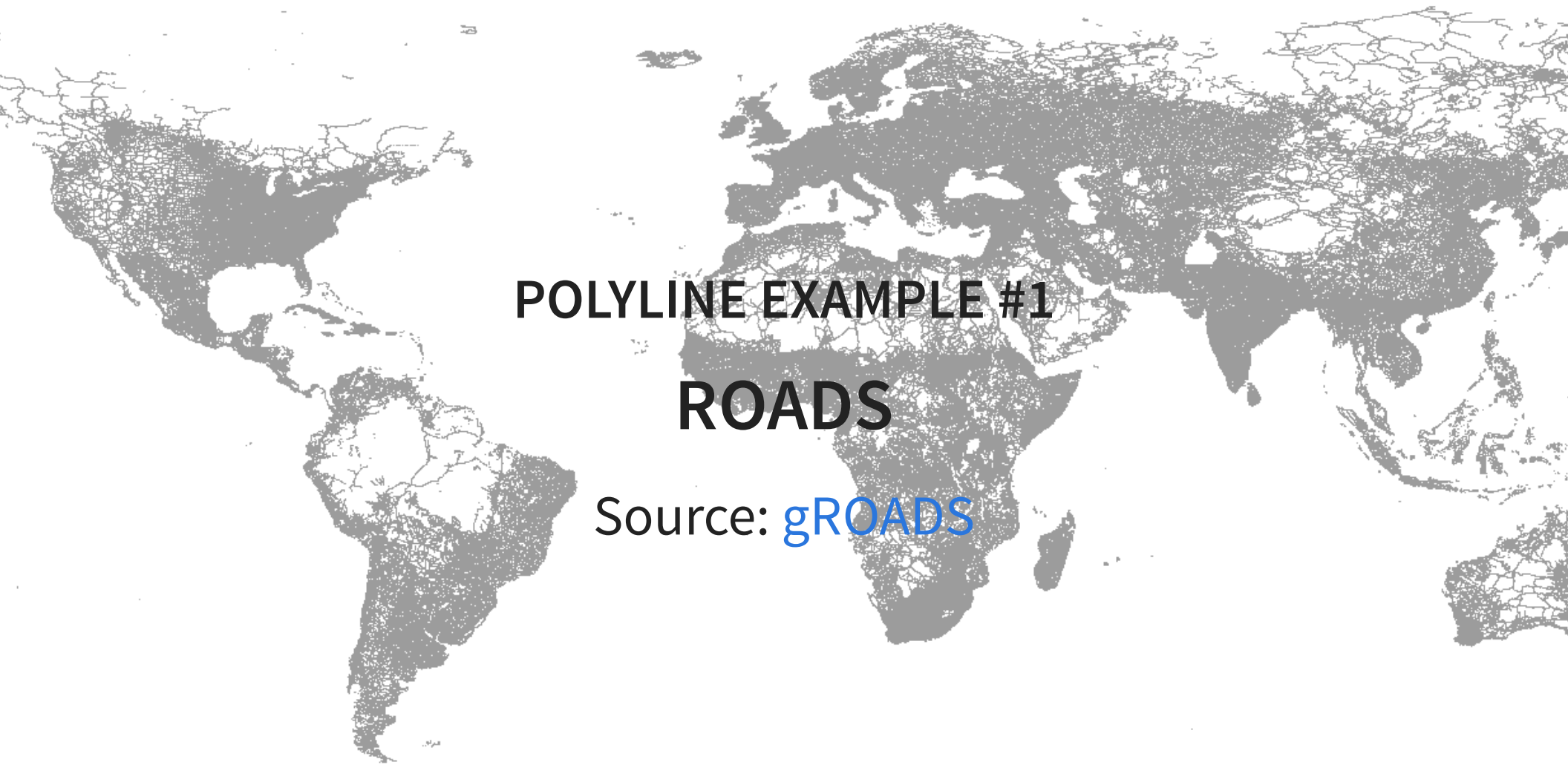
LANGUAGE GROUPS

Source: GREG

POLYGON EXAMPLE #4

LAKES & RESERVOIRS

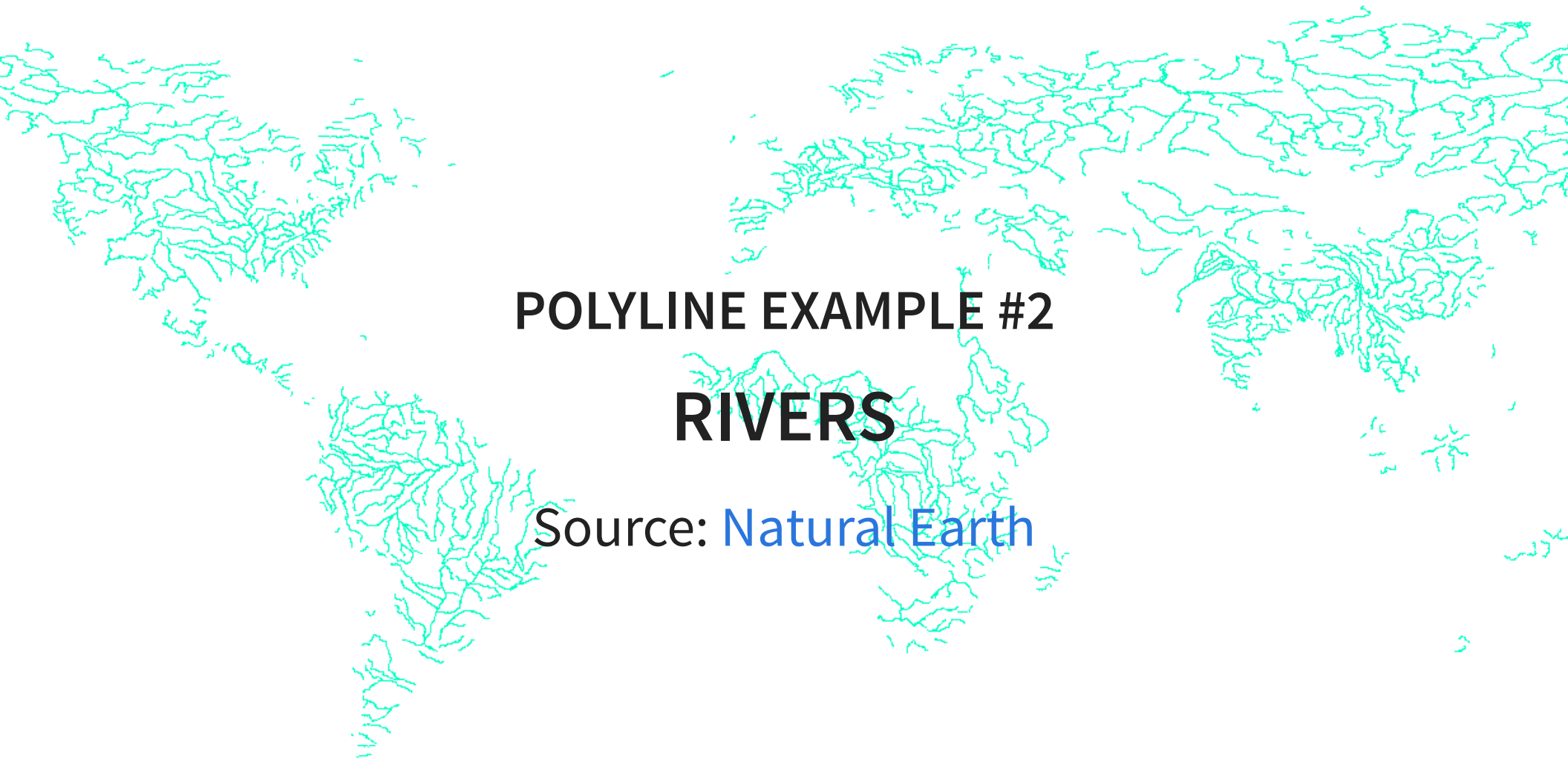
Source: Natural Earth



POLYLINE EXAMPLE #1

ROADS

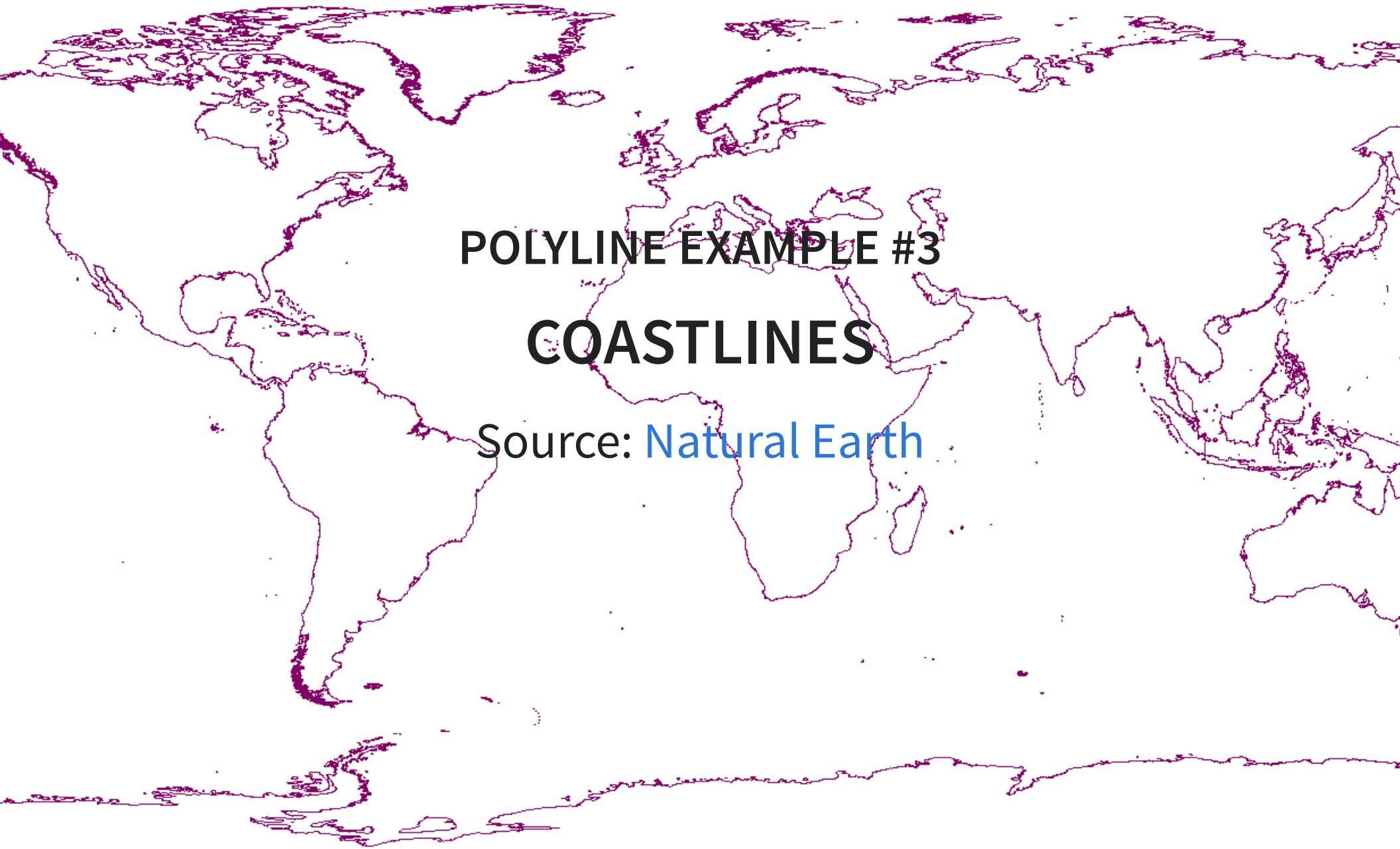
Source: gROADS



POLYLINE EXAMPLE #2

RIVERS

Source: Natural Earth


POLYLINE EXAMPLE #3
COASTLINES
Source: Natural Earth

POINT EXAMPLE

CONFLICT LOCATIONS (1997-2015)

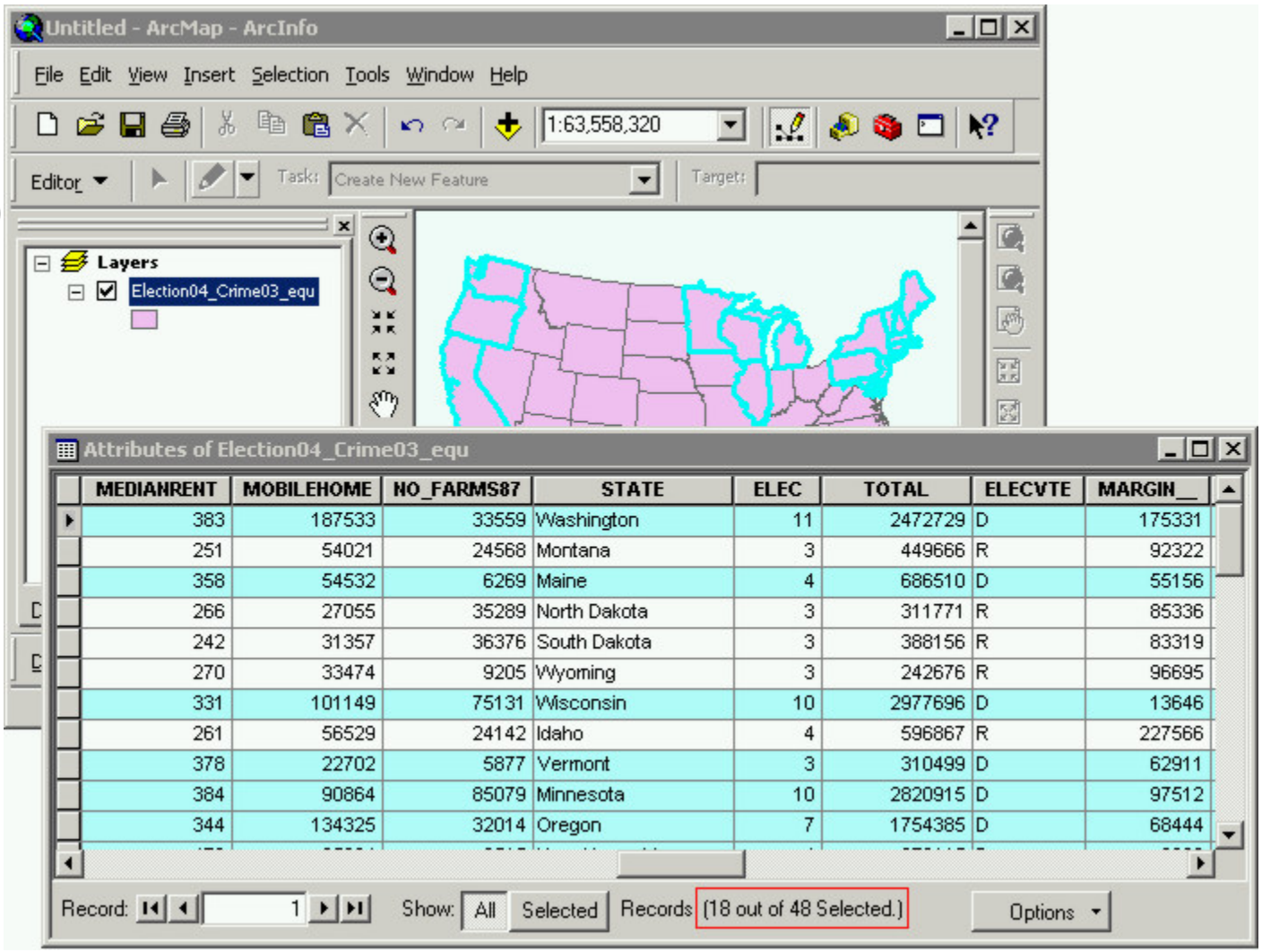
Source: ACLED

VECTOR DATA (CONT.)

Each unit: called a **feature**

Comes w/
attribute table

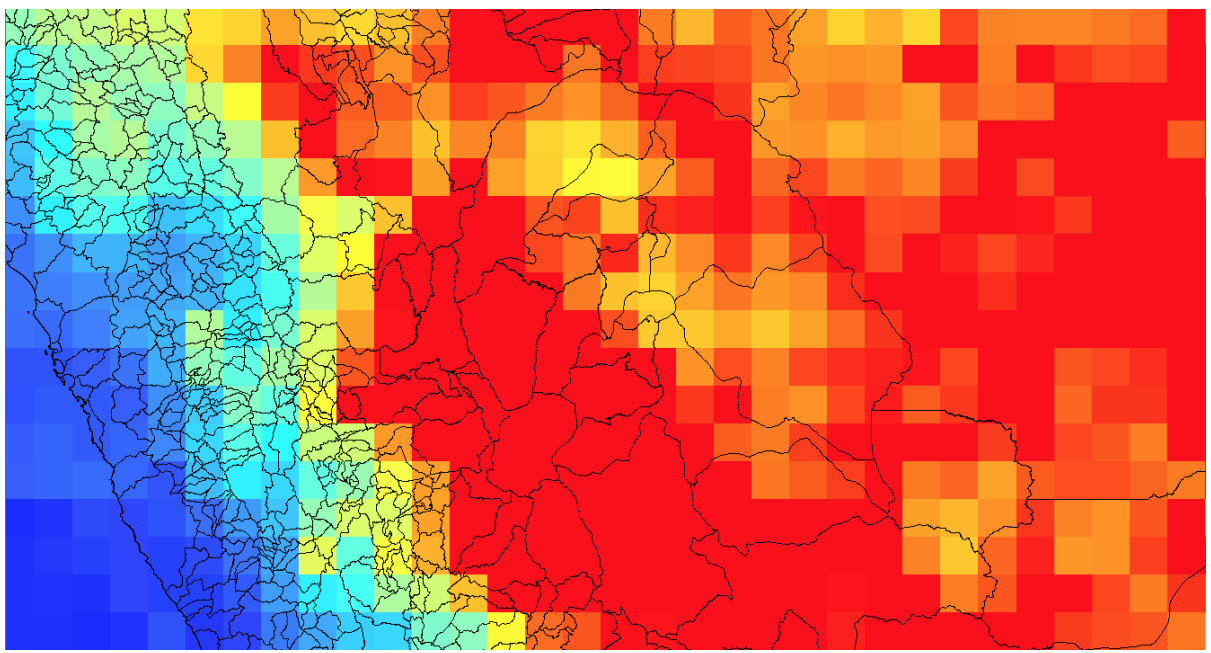
(image source)

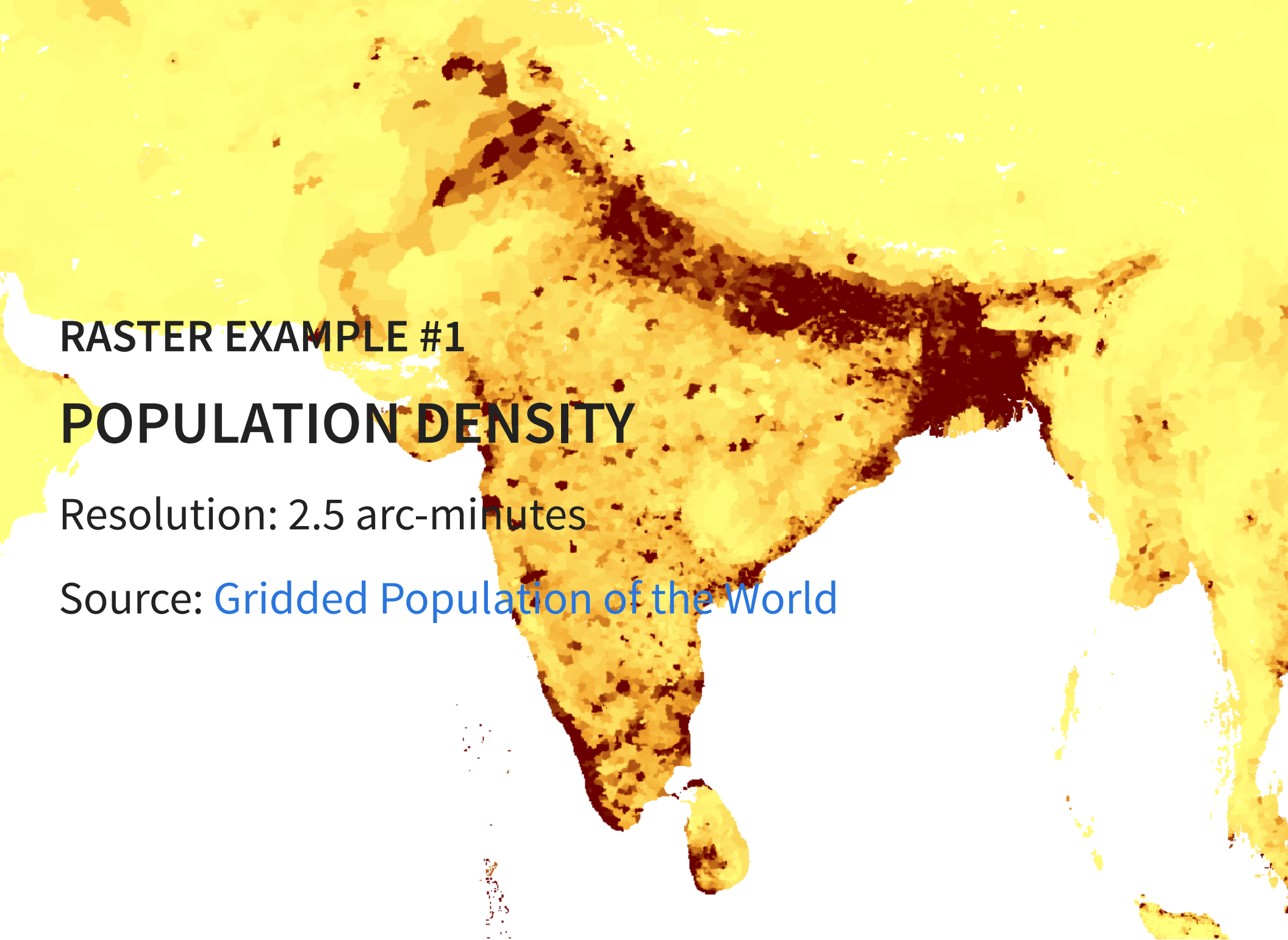


RASTER DATA

Divides the earth surface into many "square" cells (or pixels)

Each cell contains one value



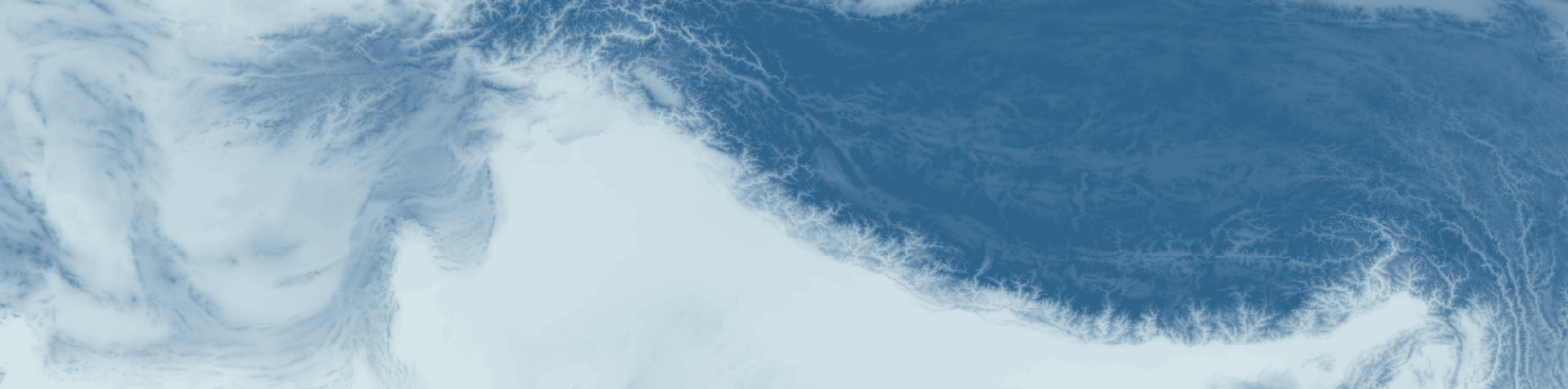


RASTER EXAMPLE #1

POPULATION DENSITY

Resolution: 2.5 arc-minutes

Source: Gridded Population of the World

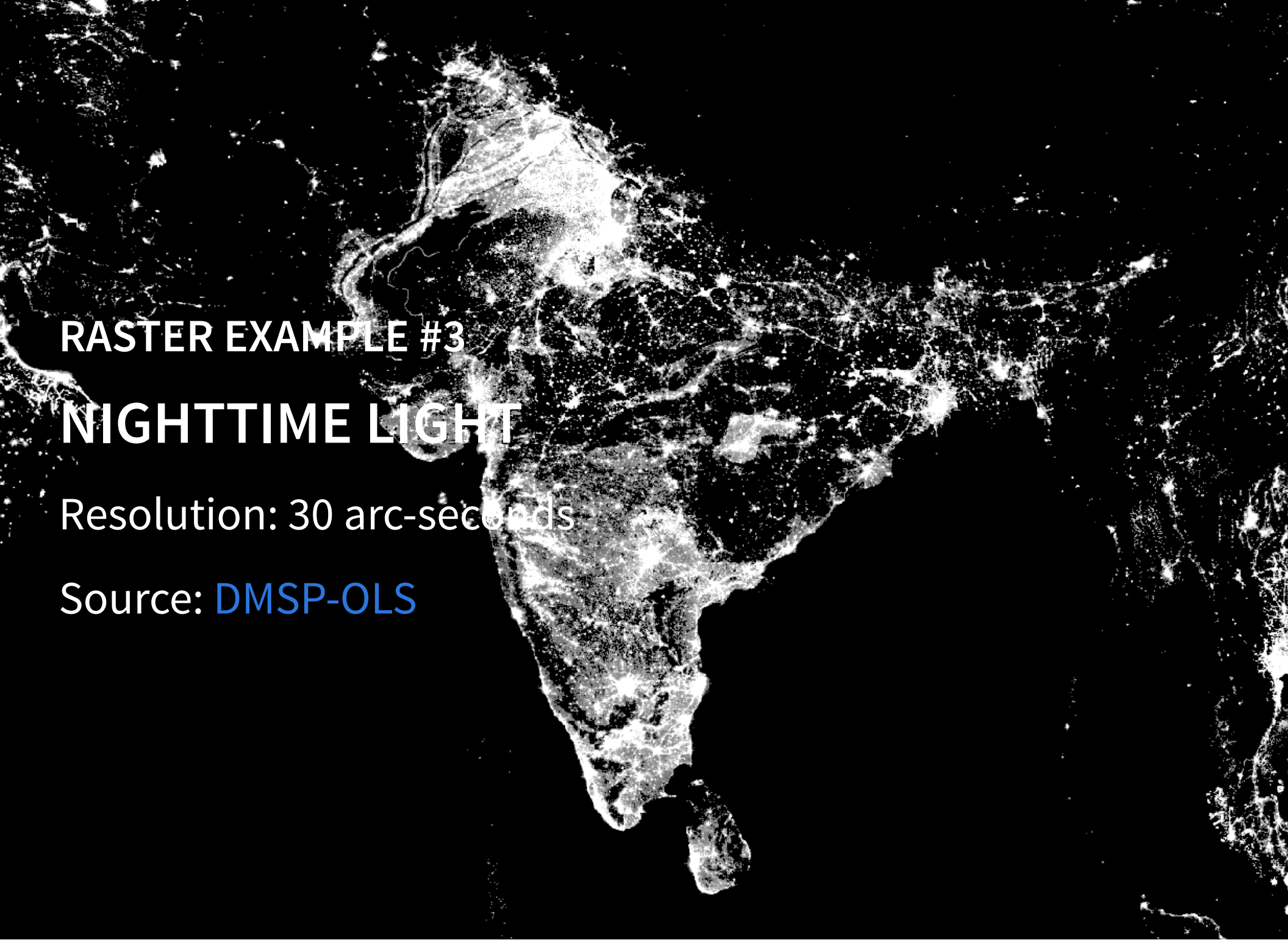


RASTER EXAMPLE #2

ELEVATION

Resolution: 30 arc-seconds

Source: STRM30


RASTER EXAMPLE #3
NIGHTTIME LIGHT
Resolution: 30 arc-seconds
Source: DMSP-OLS

1.2 COORDINATE SYSTEMS

Earth is a sphere (approximately)

Various ways to *two-dimensionally* represent Earth

Each way corresponds to a **coordinate system**

- Also called "spatial reference" or "map projection"

WHY IMPORTANT?

To merge different spatial datasets accurately

cf. Apple Map did this wrong when it was launched in 2012

WHY IMPORTANT? (CONT.)

To calculate distance and surface area properly

GEOGRAPHIC COORDINATE SYSTEMS

Each location is coded by angle from earth center

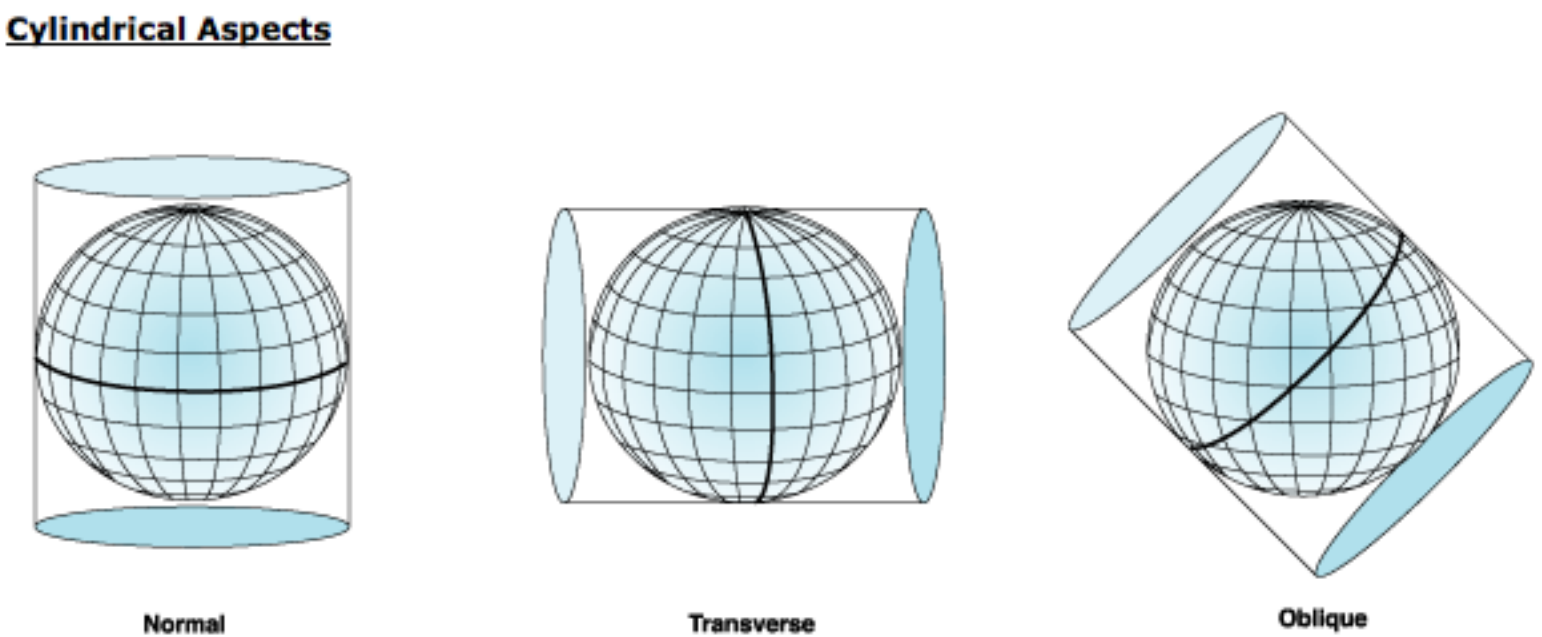
e.g. Stockholm: 59.3293° N / 18.0686° E

Most popular: **WGS 1984**

PROJECTED COORDINATE SYSTEMS

Earth surface is projected by "light" from earth center on:

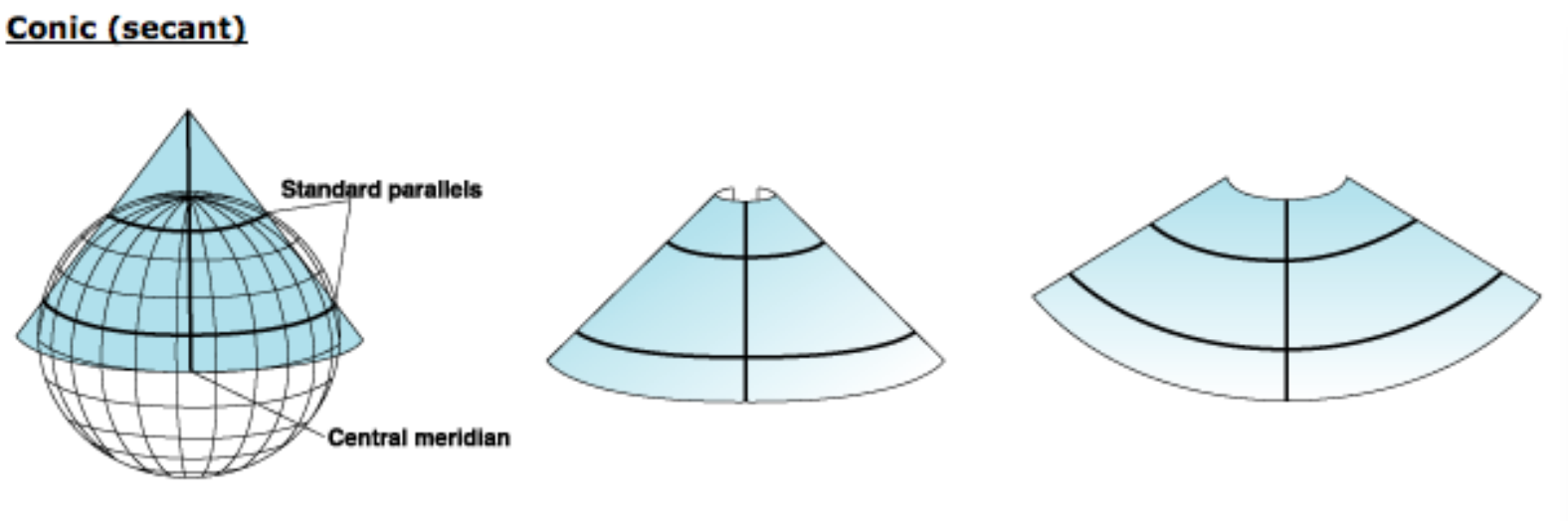
Cylinder



PROJECTED COORDINATE SYSTEMS

Earth surface is projected by "light" from earth center on:

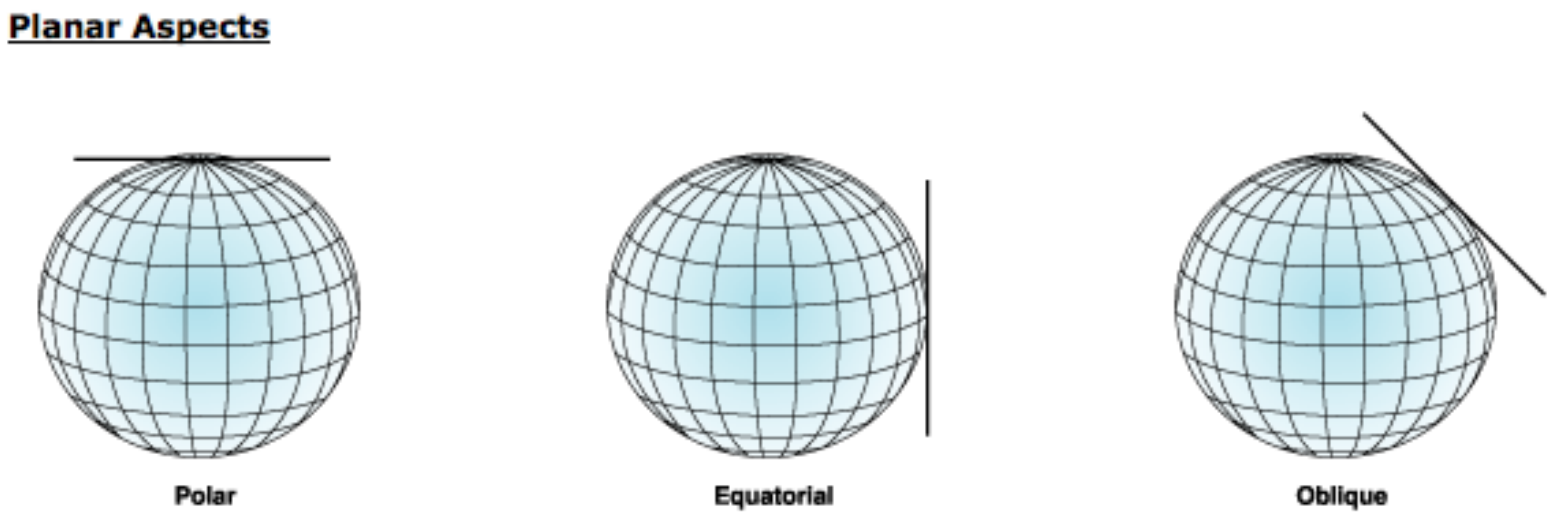
Cone



PROJECTED COORDINATE SYSTEMS

Earth surface is projected by "light" from earth center on:

Plane



PROJECTED COORDINATE SYSTEMS (CONT.)

Each location: coded in *meters* from a certain origin

village	hhn	plot	Xcoord	Ycoord
1	17	1	787877.6	644354.5
1	17	2	788041.2	644449.1
1	17	3	788041.3	644207.7
1	23	5	788942.7	644312.3
1	23	7	788942.7	644312.3
1	27	2	787733	644638.4
1	31	5	788631.2	644102.7
1	34	2	788020.9	644384.6
1	34	4	788008.4	644230.5
1	44	2	786556.3	643995.7
1	44	3	786452.8	644015.9
1	59	2	787416.6	644368.8
1	63	5	787217.3	644308.3
1	63	6	786832.4	644340.4
2	1	1	807386.6	645925.6

PROJECTED COORDINATE SYSTEMS (CONT.)

Examples (relevant for social scientists):

- UTM projections
- Equal Area projections

We will learn these projections later.

IF YOU WANT TO KNOW MORE:

Map Projections: A Working Manual, by John P. Snyder (U.S. Geological Survey, 1987) (Downloadable for free)

1.3 GIS SOFTWARE

ArcGIS

- Python-friendly
- Buggy; tricky to create map images; Windows only

QGIS

- Free; easy to create map images; compatible with any OS
- Python-unfriendly
- Tutorial: www.qgistutorials.com

⇒ ArcGIS recommended for the ease of use of Python (for replication), at least for now

1.3 GIS SOFTWARE (CONT.)

R

- Textbook: Brunsdon & Comber (2015)
- Tutorial by Nick Eubank

Geopandas

- A Python extension to work on spatial data
- Still under development (as of May 2016)

2. CREATE SPATIAL DATA ON YOUR OWN

Satellite images

Scanned old maps

Point locations

Grid cells

2.1 SATELLITE IMAGES

Images consist of pixels

Map each pixel's "color" into raster value

- By using statistical learning methods

A lot of time (and money to hire experts) needed, though.

2.1 SATELLITE IMAGE DATA (CONT.)

Some satellite images: freely available

- See "15 Free Satellite Imagery Data Sources" by GIS Geography

Examples of constructing data from satellite images

- Measuering Yields from Space
- Deforestation

APPLICATION: BURGESS ET AL. 2012

of districts w/i province ↑ ⇒ Deforestation ↑

Theory:

- Each district govt official engages in Cournot competition in selling (illegal) logging permits
- More districts ⇒ More supply of illegal permits

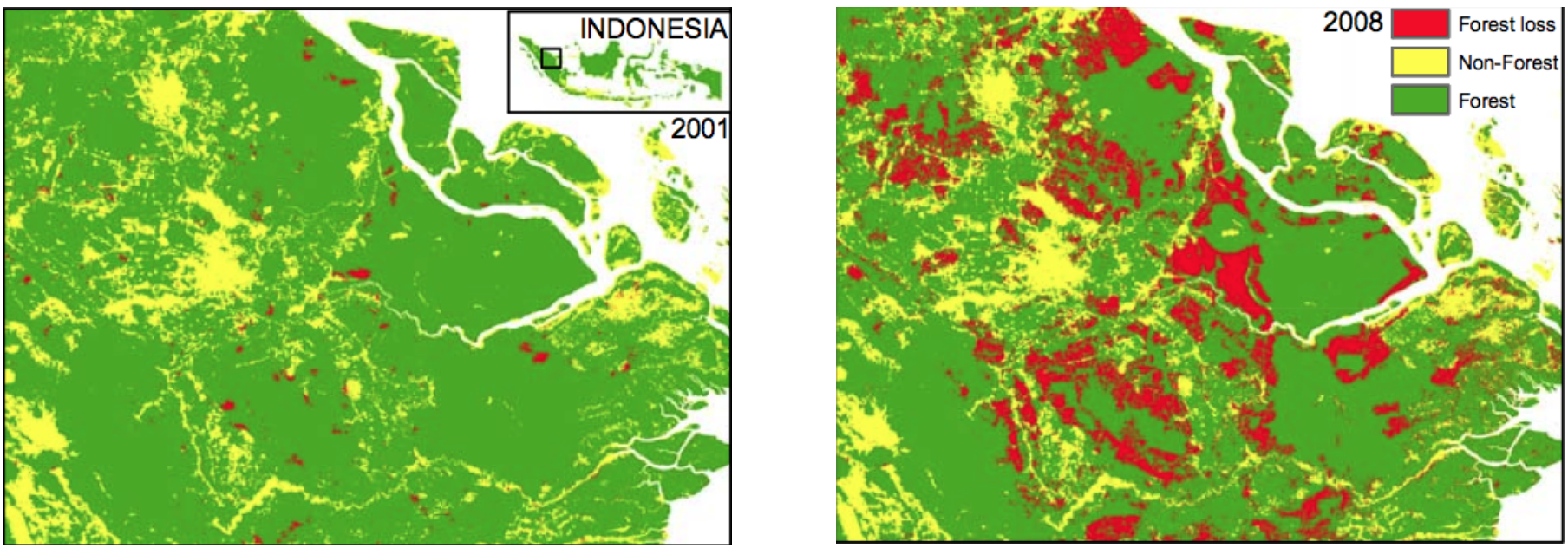
APPLICATION: BURGESS ET AL. 2012 (CONT.)

Cannot rely on official stats of logging

⇒ Use satellite images

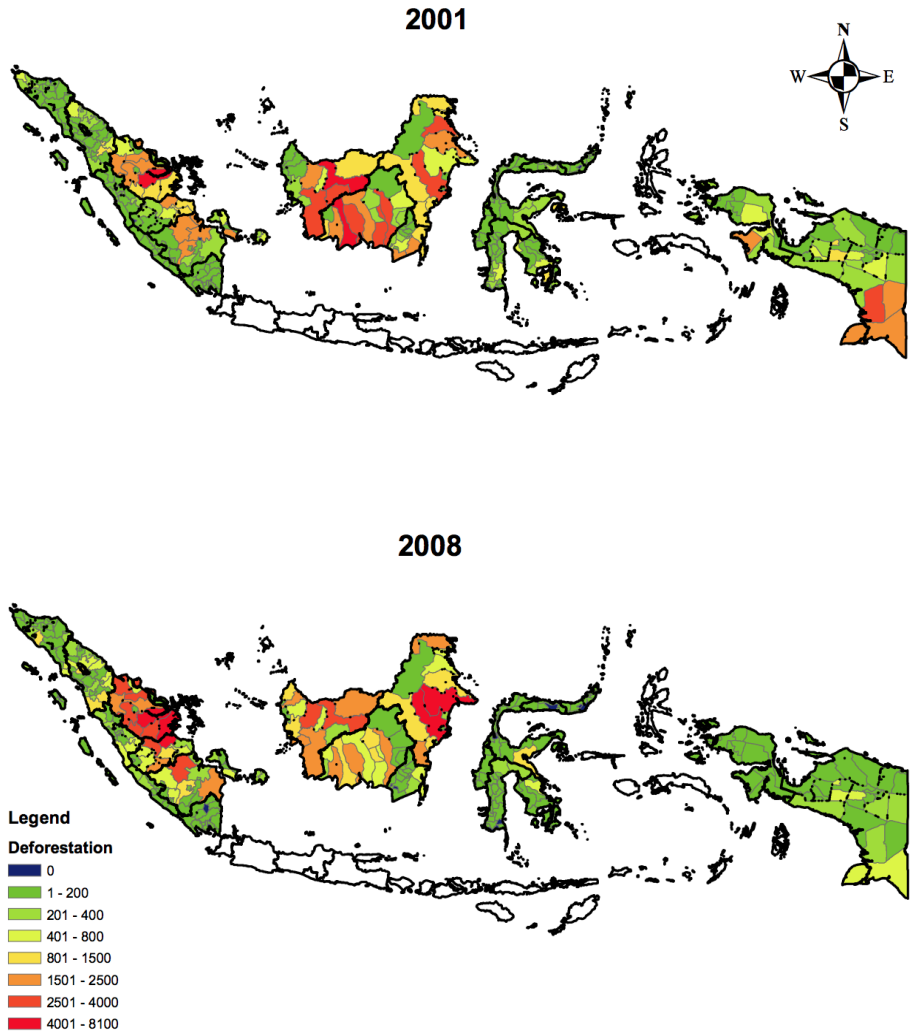
- Spatial resolution: 250m x 250m pixel
- Data: electromagnetic radiation strength in 36 bands of spectrum
- Develop algorithm to convert radiation patterns to forest coverage

PIXEL-LEVEL DATA ON DEFORESTATION



(Figure I of Burgess et al. 2012)

DISTRICT-LEVEL DATA ON DEFORESTATION



(Figure II of Burgess et al. 2012)

2.2 SCANNED OLD MAPS

First, geo-reference the map

- See Yale Map Collection (2009) (pp. 8-10)

Then, create vector data by tracing lines w/ mouse

- See ArcGIS 10: Editing & Creating Your Own Shapefiles (Parts 3-6)

Also time-consuming but feasible with patience

APPLICATION: BURGESS ET AL. 2015

Did Kenyan presidents build more roads for their co-ethnics?

Digitize Michelin maps for Kenya since 1961

Track road network expansion over time

DIGITIZING OLD MAPS

Michelin map in 1961

Digitization and Standardization in GIS

(source: Remi Jedwab's presentation slide)

2.3 POINTS

First, create a table in text format, where:

- Each row: location
- Column 1: longitude (x value)
- Column 2: latitude (y value)
- Other columns: attributes of each location
  - Name
  - Statistics
  - Key (unique ID)
  - Foreign keys (for merging with other data)

HOW TO OBTAIN LONGITUDE & LATITUDE?

GPS receiver

- If you conduct your own survey

Online gazetteer

- If location names are available, search at:
  - Geonames
    - Geonames Tools toolbox automates search
  - Global Gazetteer Version 2.3
  - JRC Fuzzy Gazetteer
- If address is available, use Google Geocoder
  - Stata ado `geocode3` automates search

2.3 POINTS (CONT.)

To convert the text file table into point vector data in ArcGIS, use:

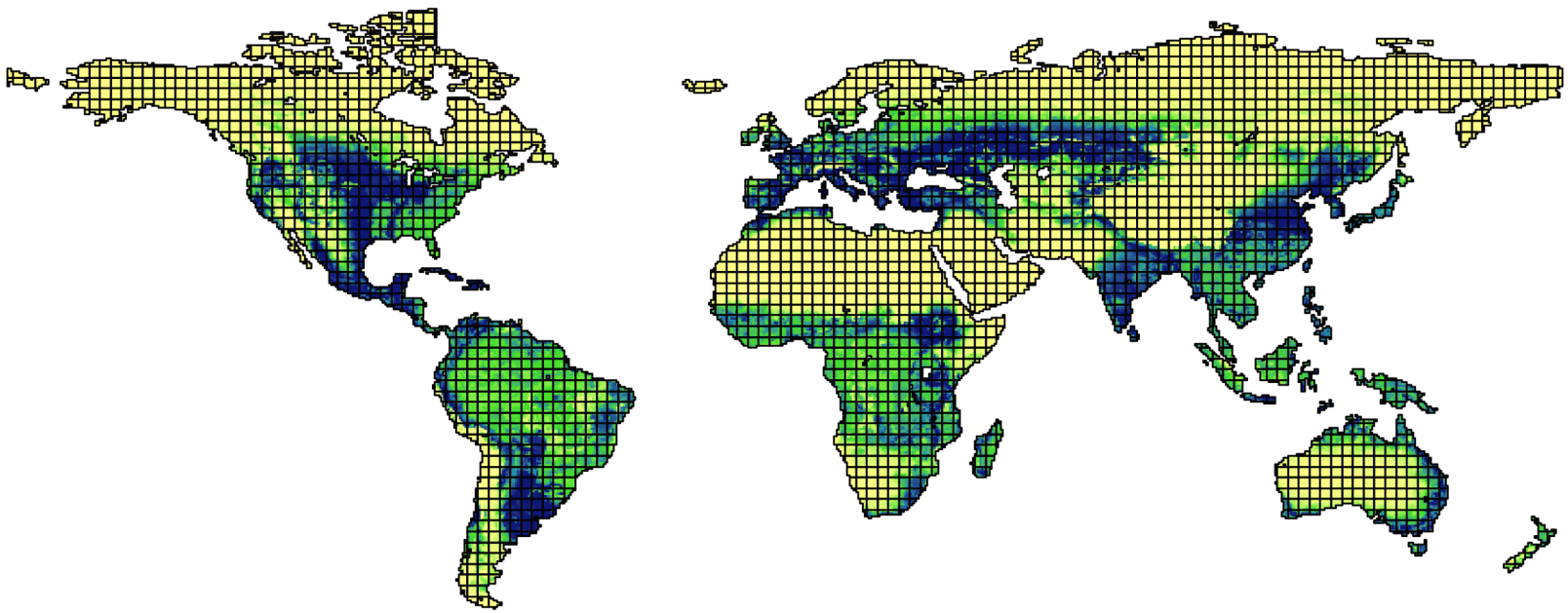
- Make XY Event Layer
- Copy Features

These are the examples of **geo-processing tools**

PYTHON CODE FOR CREATING POINT FEATURES

```
import arcpy

input_table = "coordiates.txt"
output_shp = "points.shp"

varname_x = "longitude"
varname_y = "latitude"

coordinate_system = arcpy.SpatialReference(4326)

arcpy.MakeXYEventLayer_management(
        input_table, varname_x, varname_y,
        "Layer", coordinate_system)
arcpy.CopyFeatures_management("Layer", output_shp)
```

2.4 GRID CELLS



Any size of grid cell polygons can be created in ArcGIS

2.4 GRID CELLS (CONT.)

Useful for:

- Merging weather data
  - WorldClim (Dell, Jones, & Olken 2009)
  - GCPC (Miguel et al. 2004)
  - TOMS air pollution index (Jayachandran 2009)
- Exogenous boundaries

2.4 GRID CELLS (CONT.)

To create grid cell polygons in ArcGIS, use:

- Create Fishnet
- Define Projection

These are also the examples of **geo-processing tools**

PYTHON CODE FOR CREATING POINT FEATURES

```
import arcpy

output_shp = "gridcells25.shp"
cellsize = "2.5"
bottom_left = "-180 -65"
top_right = "180 85"
y_axis = "-180 -55"
coordinate_system = arcpy.SpatialReference(4326)

arcpy.CreateFishnet_management(
        output_shp, bottom_left, y_axis,
        cellsize, cellsize, "0", "0", top_right,
        "NO_LABELS", "", "POLYGON")
arcpy.DefineProjection_management(
        output_shp, coordinate_system)
```

GEO-PROCESSING TOOLS

Master ArcGIS = Know which geo-processing tools to use

Takes vector/raster data as inputs

Most will create new vector/raster data

- Some tools just overwrite the input data

Can be executed in Python

- Should be, for replication

PLAN FOR REST OF THIS LECTURE

Introduce each geo-processing tool

Demonstrate how it's used by economists

3. MERGE SPATIAL DATASETS

1. Spatial Join
2. Intersect + Dissolve
3. Zonal Statistics as Table

3.1 SPATIAL JOIN

Add new variables from a second vector data

Based on **location**

- Not on key variables as in Stata's `merge`

3.1 SPATIAL JOIN (CONT.)

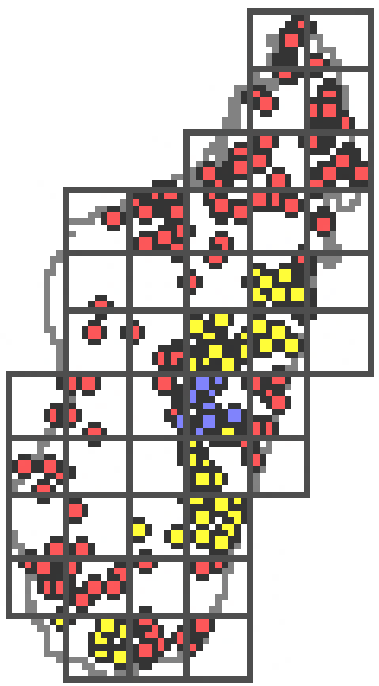
One useful application: merge with weather data

Weather data: available at grid cell level

- No information on country, province, district etc.

3.1 SPATIAL JOIN (CONT.)

⇒ Spatial Join specifies which grid cells are relevant for each observation



PYTHON CODE FOR SPATIAL JOIN

```
import arcpy

target = "cities.shp"
join = "weather_data_cells.shp"
output = "cities_with_weather_data.shp"

arcpy.SpatialJoin_analysis(
        target, join, output)
```

APPLICATION 1: FEYRER & SACERDOTE (2009)

European colonization ⇒ Economic development?

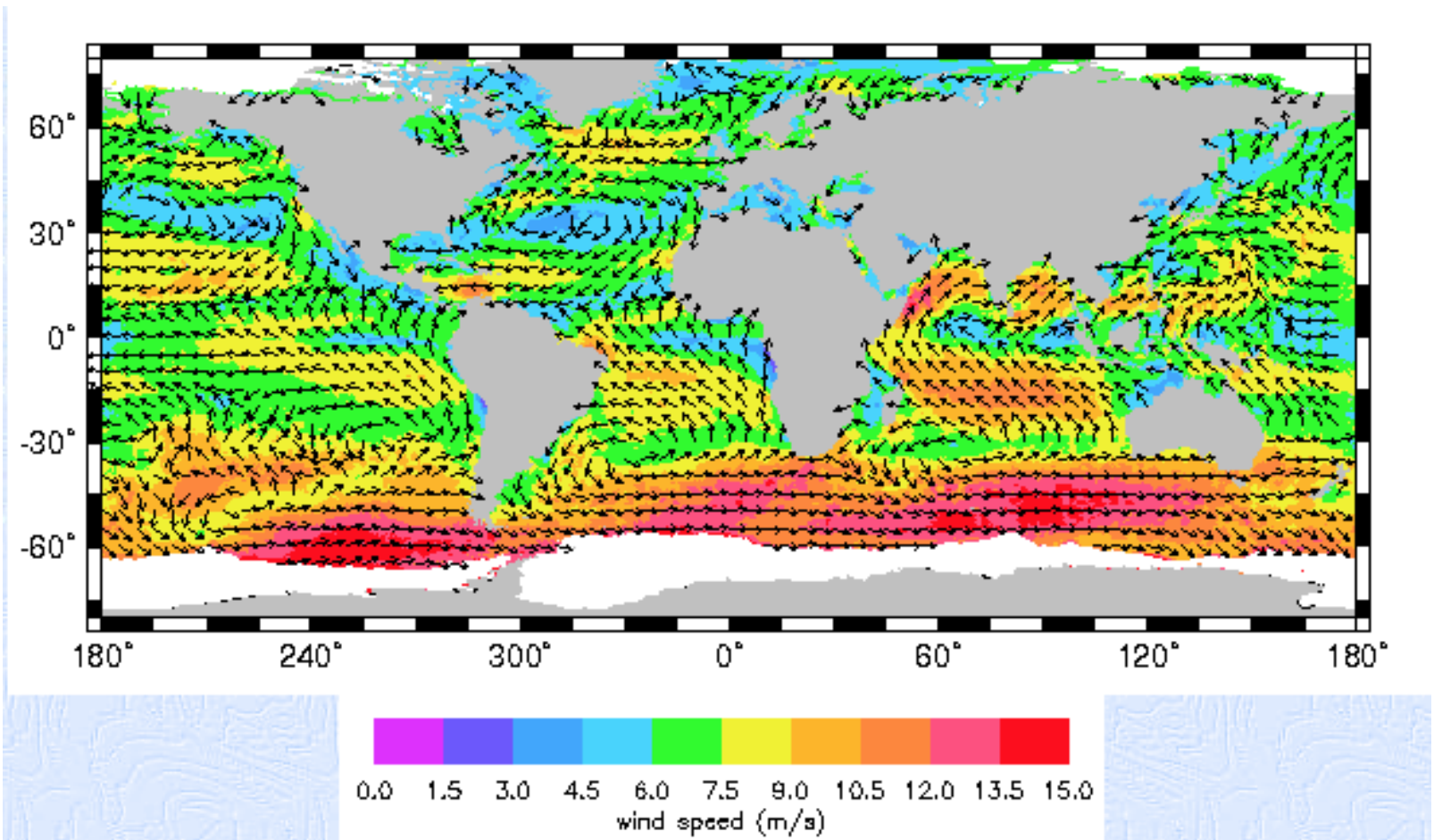
Sample: islands (geo-referenced)

IVs for duration of colonization:

- Mean east-west wind speed
- SD of east-west wind speed

APPLICATION 1: FEYER & SACERDOTE (2009) (CONT.)

Wind data: CERSAT (1° x 1°)



APPLICATION 2: ALSAN (2015)

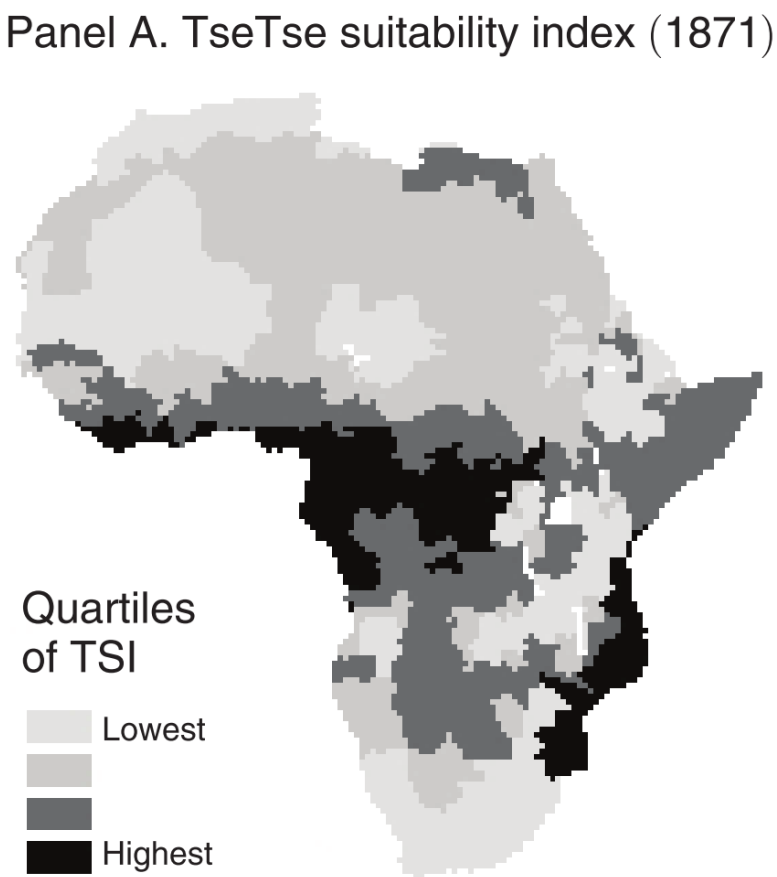
Tsetse flies ⇒ Africa's underdevelopment?

Weather in 1871 at 2° x 2° resolution

Temperature & humidity fed into a model to predict Tsetse fly survival

APPLICATION 2: ALSAN (2015) (CONT.)

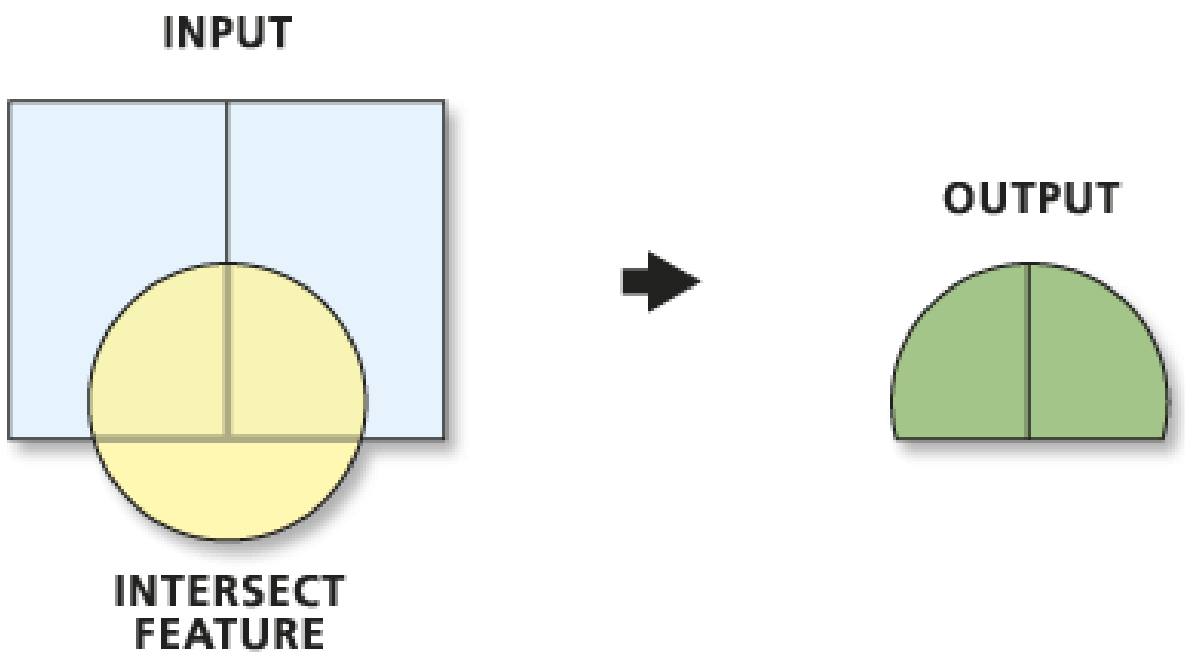
Matched with Ethnographic Atlas



(Figures 5A and 3A of Alsan 2015)

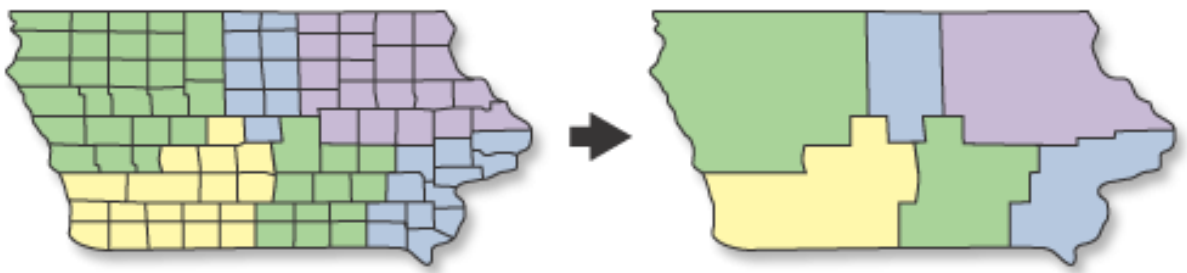
3.2 INTERSECT + DISSOLVE

Intersect: Creates intersection features



3.2 INTERSECT + DISSOLVE (CONT.)

Dissolve: Combines features by key variables



Can also create summary statistics

- Stata's `collapse`

3.2 INTERSECT + DISSOLVE (CONT.)

Can calculate # of polygons/polylines w/i zone polygon

```
import arcpy

inFeatures = ["counties.shp", "streams.shp"]
intersectOutput = "streams_by_country.shp"
dissolve_field = "county_id"
outFeatures = "counties_number_of_streams.shp"

arcpy.Intersect_analysis(
        inFeatures, intersectOutput)
arcpy.Dissolve_management(
        intersectOutput, outFeatures,
        dissolve_field, "COUNT")
```

APPLICATION 1: HOXBY (2000)

Competition ⇒ School quality ↑?

IV: **# of streams** w/i city

- More streams ⇒ More school districts

APPLICATION 2: BAI AND JIA (2016)

In early 20c, Imperial China abolished 1300-year-old civil service exams

Prefectures w/ higher quota ⇒ more uprisings during the 1911 Revolution

IV: **# of streams** w/i prefecture

- Quota depends on # of counties w/i prefecture

3.3 ZONAL STATISTICS AS TABLE

Calculates sum stat of raster values w/i zone

Zone: defined by polygon or polyline

- Stata's `collapse` by zone, executed on raster data

3.3 ZONAL STATISTICS AS TABLE (CONT.)

- Mean / Standard deviation
- Min / Max / Range
- Sum
- Count

For integer raster:

- Median
- Variety (# of unique values)
- Majority (most frequent value)
- Minority (least frequent value)

3.3 ZONAL STATISTICS AS TABLE (CONT.)

If unit of analysis is point, use either:

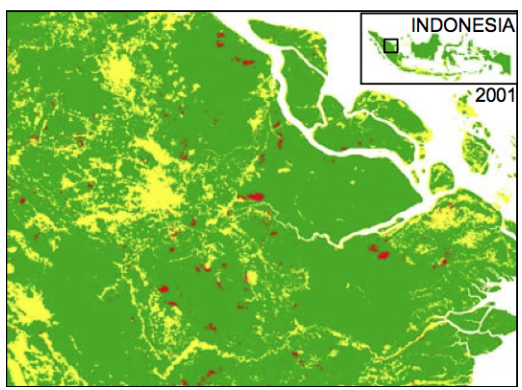
- **Extract Multi Values To Points**
- **Buffer** + Zonal Statistics as Table

PYTHON CODE FOR ZONAL STATISTICS

```
import arcpy
arcpy.CheckOutExtension("spatial")

# Inputs
elevation = "srtm30.dem"
districts = "district.shp"
# Intermediate
zonalstat = "zonalstat.dbf"
# Outputs
mean_elevation = "mean_elevation.xls"

arcpy.gp.ZonalStatisticsAsTable_sa(
        districts, "dist_id", elevation,
        zonalstat, "DATA", "MEAN")
arcpy.TableToExcel_conversion(
        zonalstat, mean_slope)
```

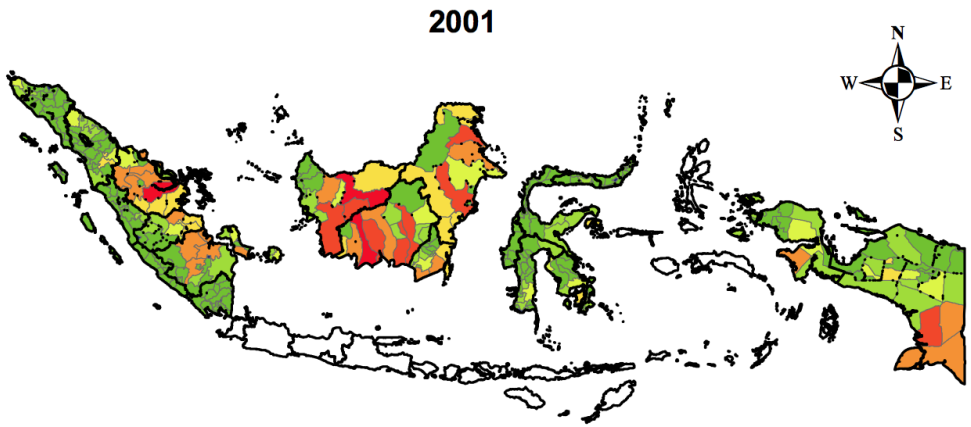
APPLICATION 1: BURGESS ET AL. 2012

Pixel-level



⇉

District-level



APPLICATION 2: NIGHTTIME LIGHT STUDIES

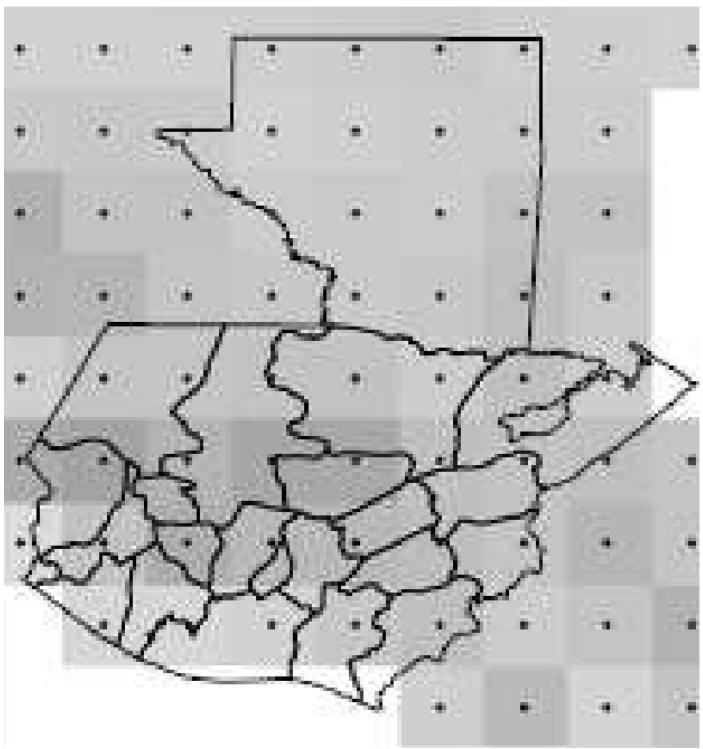
Obtain the mean cell value by:

- Country (Henderson et al. 2012)
- Provinces (Hodler & Raschky 2014)
- Electoral districts (Baskaran et al. 2015)
- Ethnic homelands (Michalopoulos & Papaioannou 2013 / 2014, Alesina et al. 2016)

APPLICATION 3: DELL ET AL. 2012

Temperature ↑ ⇒ Economic growth ↓?

Use population-weighted average temperature



(Figure 2.3 of Dell 2009)

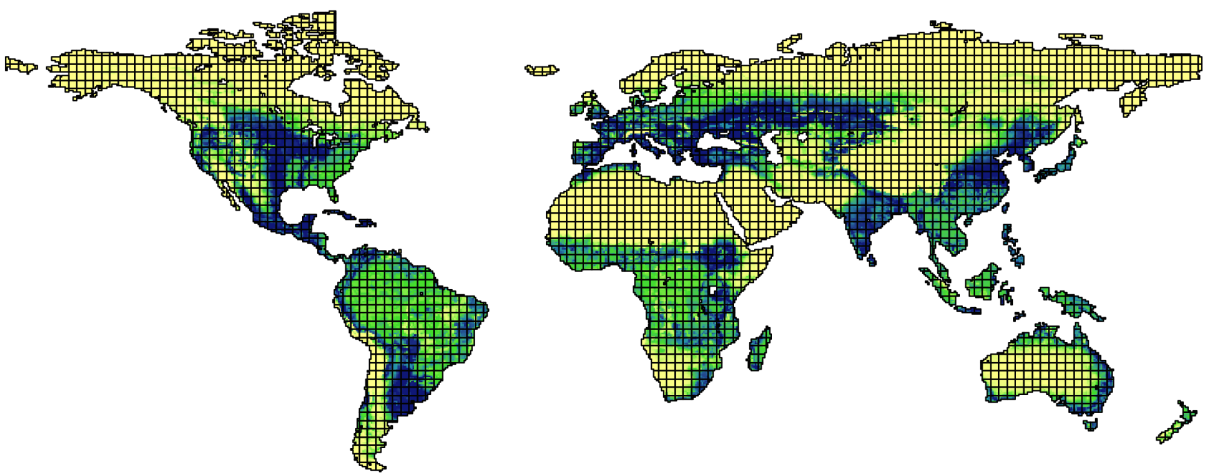
APPLICATION 4: MICHALOPOULOS (2012)

Geographic diversity ⇒ Ethnic diversity?

Empirical challenge:

- Endogeneity of geographic diversity by country formation

⇒ 2.5°x2.5° grid cells as units of analysis



APPLICATION 4: MICHALOPOULOS (2012) (CONT.)

How to measure ethnic & geographic diversity, then?

ArcGIS helps:

- Intersect + Dissolve ⇒ # of languages spoken
- Zonal Statistics ⇒ S.D. of elevation / land quality

APPLICATION 4: MICHALOPOULOS (2012) (CONT.)

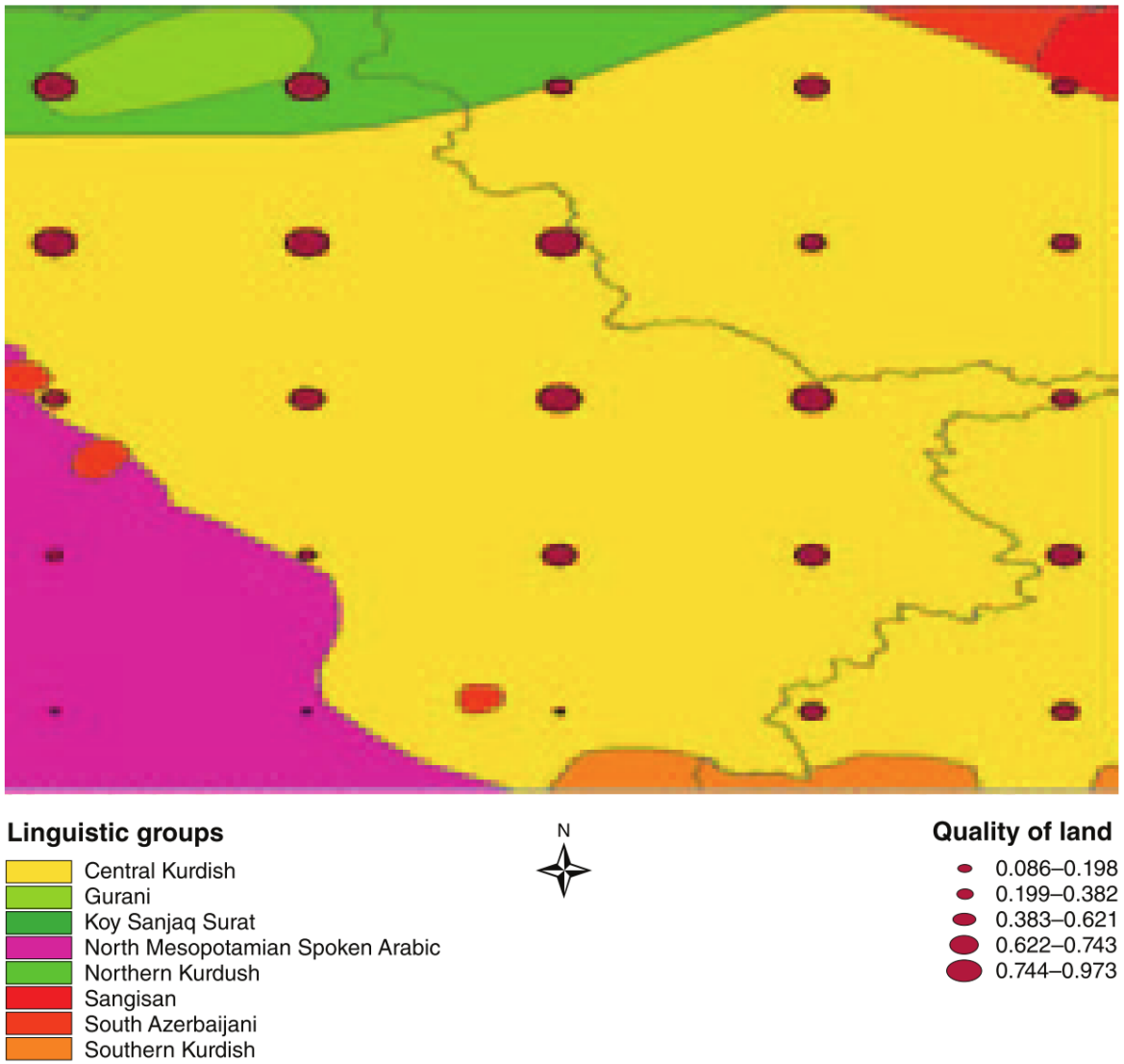


FIGURE 7. EXAMPLE OF A VIRTUAL COUNTRY

4. ELEVATION

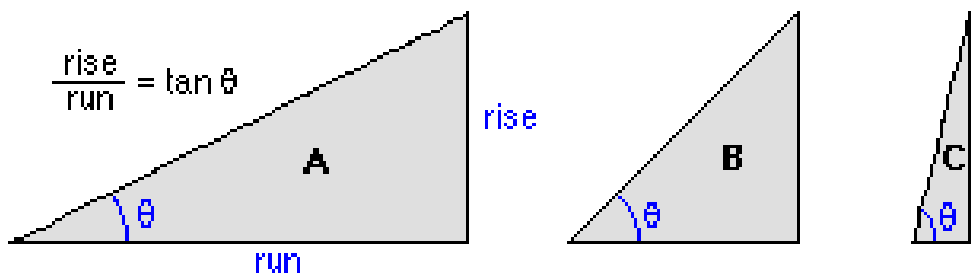
1. Slope
2. Slope + Reclassify
3. Irregular Terrain Model

4.1 SLOPE

Returns either θ or $\tan\theta$ in diagram below

Degree of slope = θ

Percent of slope = $\frac{\text{rise}}{\text{run}} * 100$



Degree of slope =	30	45	76
Percent of slope =	58	100	373

4.1 SLOPE (CONT.)

$\tan\theta$ for cell e is obtained by

a	b	c
d	e	f
g	h	i

$$\Rightarrow \quad \tan\theta = \sqrt{(dz/dx)^2 + (dz/dy)^2}$$

where

$$\frac{dz}{dx} = \left[\frac{c + 2f + i}{4} - \frac{a + 2d + g}{4}\right]/2$$

$$\frac{dz}{dy} = \left[\frac{a + 2b + c}{4} - \frac{g + 2h + i}{4}\right]/2$$

APPLICATION 1: DINKELMAN (2011)

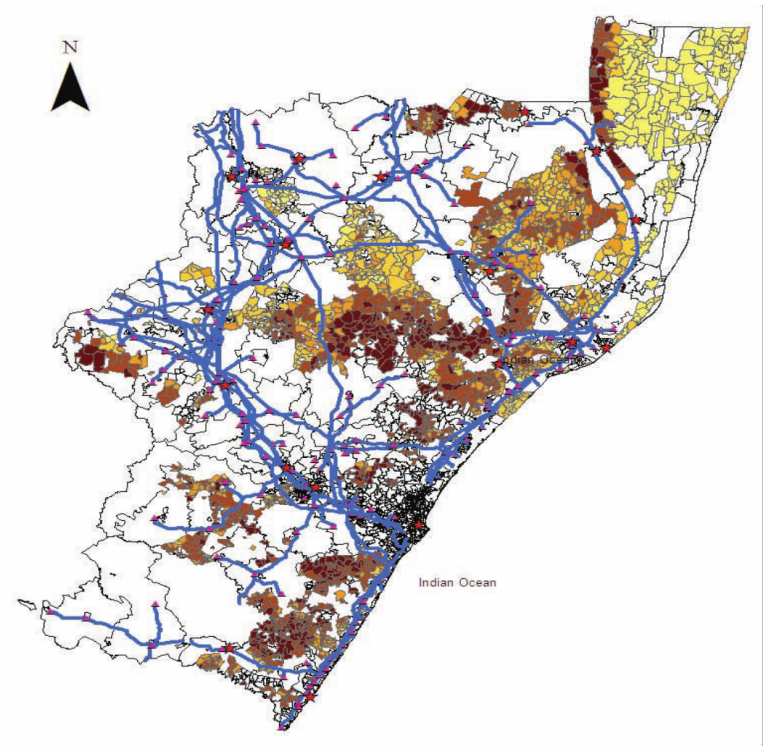
Electrification in South Africa (1996-2001)

$\Rightarrow$ Female labor supply $\uparrow$

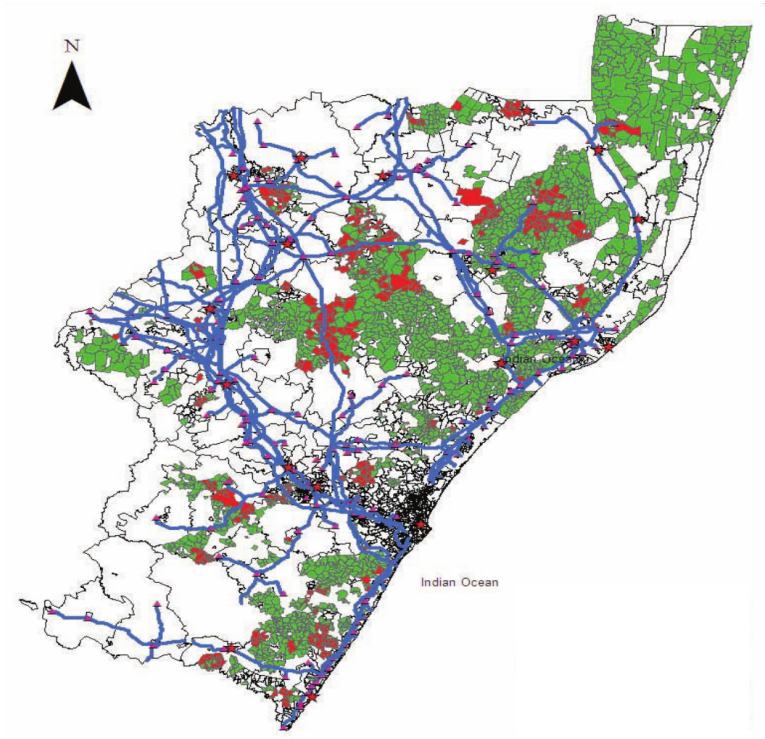
IV: **mean land slope**

- Flat terrain: cheap to lay power lines

APPLICATION 1: DINKELMAN (2011) (CONT.)



Slope (lighter = flatter)



Electrification (red)

APPLICATION 2: QIAN (2008)

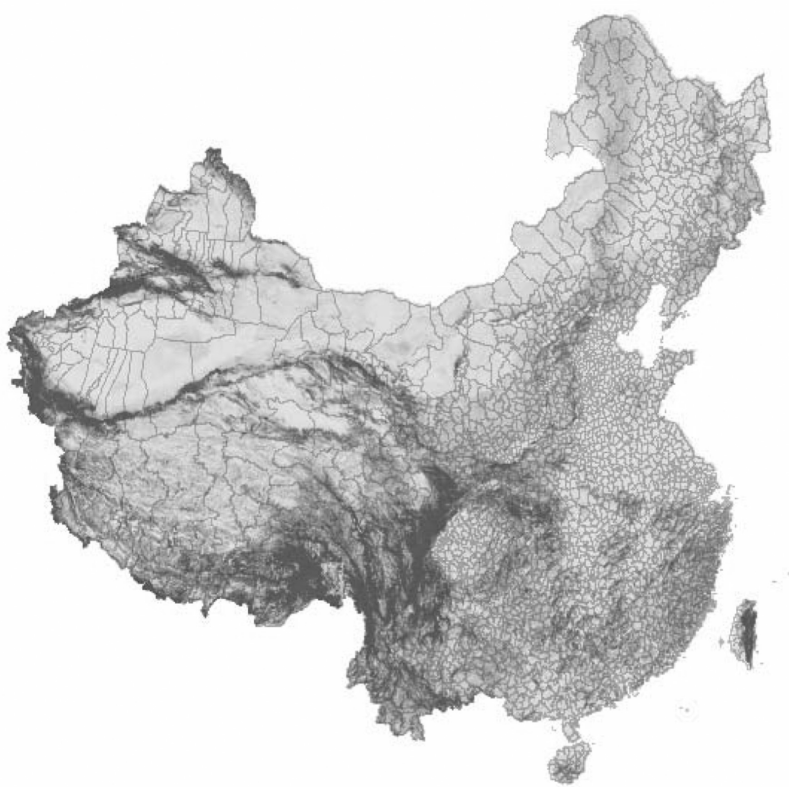
Tea production ↑ in China due to liberalization in 1979

⇒ Male-to-female ratio ↓

IV: **mean land slope**

- Tea grows in hilly terrain

APPLICATION 2: QIAN (2008) (CONT.)



Slope (darker = steeper)

Tea (darker = more)

PYTHON CODE FOR AVERAGE SLOPE

```
import arcpy
arcpy.CheckOutExtension("spatial")

# Inputs
elevation = "srtm30.dem"
districts = "district.shp"
# Intermediates
slope = "slope.tif"
zonalstat = "zonalstat.dbf"

arcpy.gp.Slope_sa(
        elevation, slope, "PERCENT_RISE", "0,000009")
arcpy.gp.ZonalStatisticsAsTable_sa(
        districts, "dist_id", slope,
        zonalstat, "DATA", "MEAN")
```

4.2 SLOPE + RECLASSIFY

Reclassify: Creates categorical raster data

Example: a dummy variable for slope 3-6%

Old values	New values
0 - 3	0
3 - 6	1
6 - 193.229706	0
NoData	NoData

PYTHON CODE FOR SLOPE CATEGORY

```
import arcpy
arcpy.CheckOutExtension("spatial")

# Inputs
elevation = "srtm30.dem"
# Outputs
slope = "slope.tif"
slope_3_6 = "slope3to6.tif"

arcpy.gp.Slope_sa(
        elevation, slope, "PERCENT_RISE", "0,000009")
arcpy.gp.Reclassify_sa(
        slope, "Value",
        "0 3 0;3 6 1;6 193,22999999999999 0",
        slope_3_6, "DATA")
```

APPLICATION 1: DUFLO & PANDE (2007)

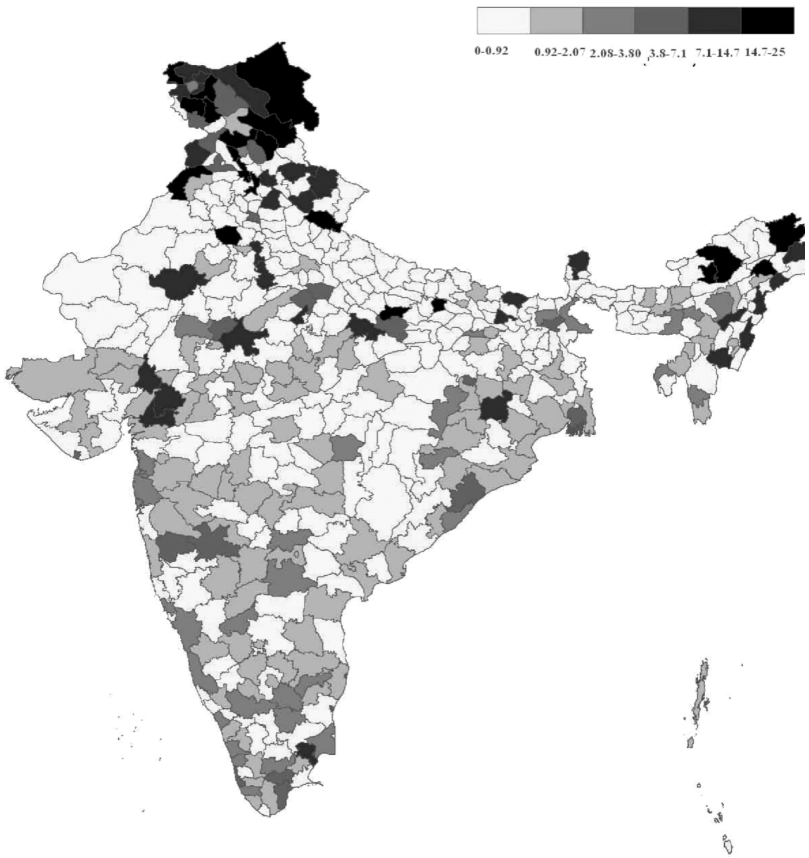
Irrigation dams $\Rightarrow$ Poverty $\downarrow$?

IV: Fraction of river areas in three slope ranges

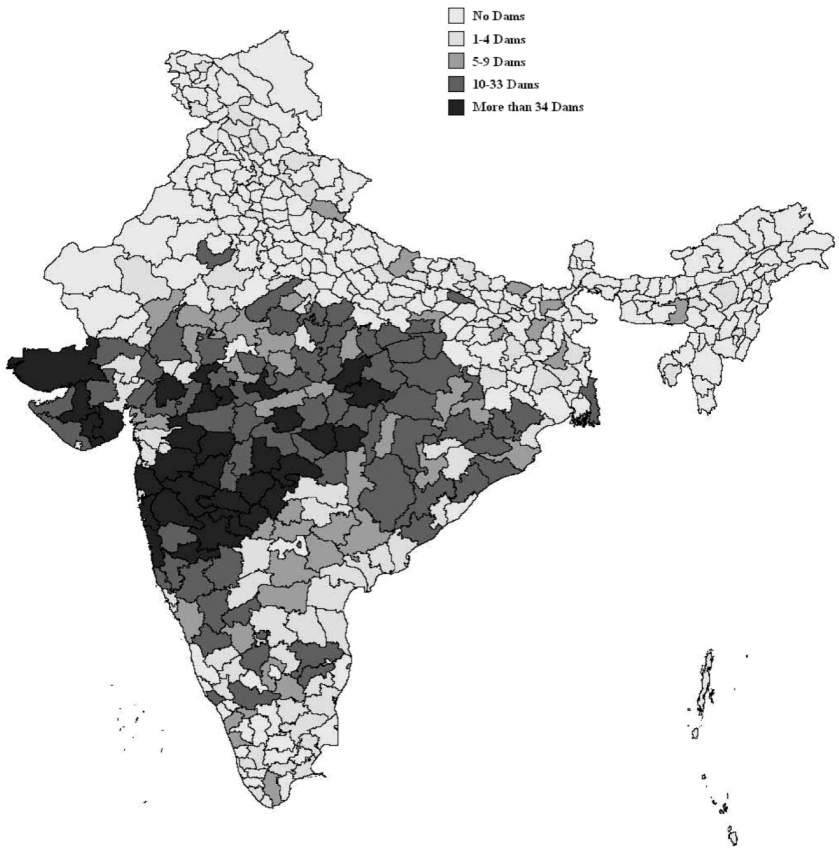
$\Leftarrow$ Easy to build if river slope is:

- Moderate (1.5-3%) for irrigation dams
- Very steep (6%+) for hydroelectricity dams

APPLICATION 1: DUFLO & PANDE (2007) (CONT.)



River slope (darker = steeper)



Dams (darker = more)

APPLICATION 1: DUFLO & PANDE (2007) (CONT.)

Intersect + Dissolve ⇒ River by districts

Slope + Reclassify ⇒ Indicator for each slope range

Zonal Statistics as Table ⇒ Fraction of river areas in each slope range by district

APPLICATION 2: SAIZ (2010)

Measures % of areas with slope > 15% as unsuitability for urban development

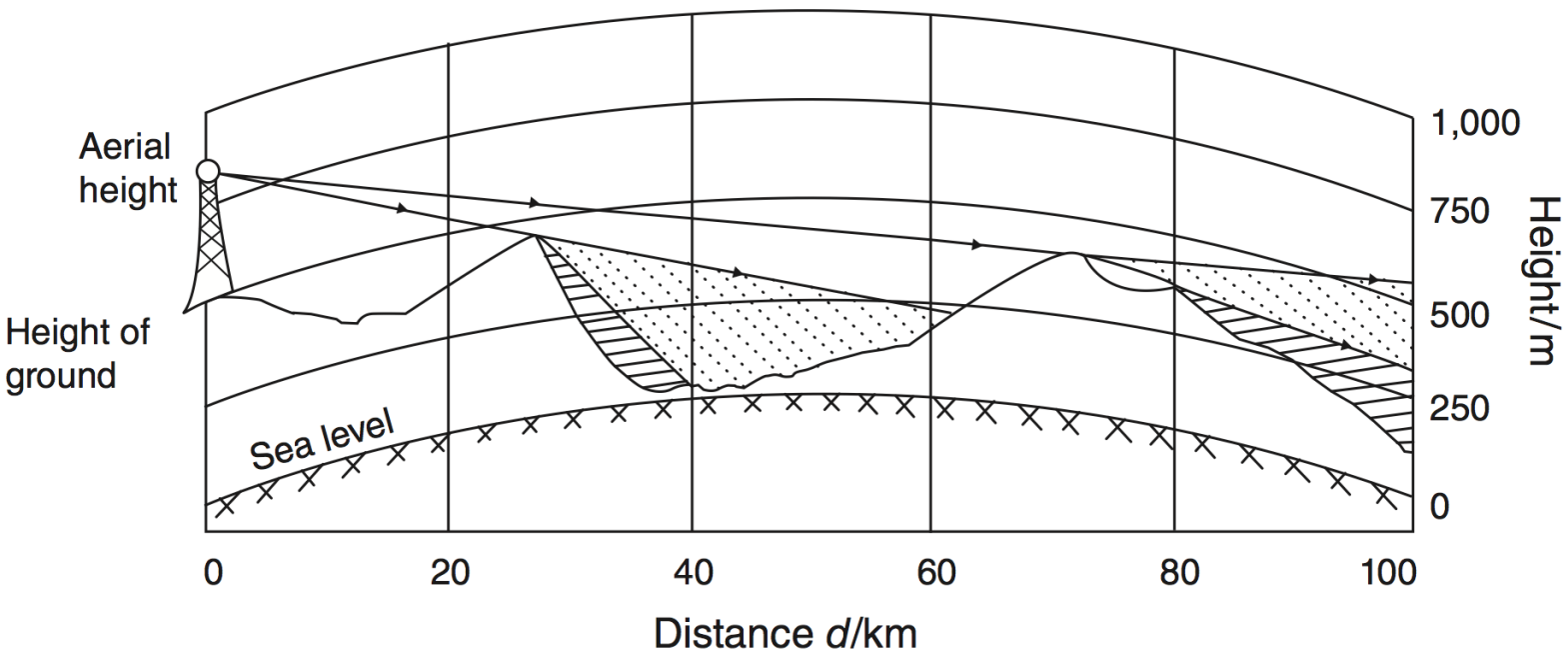
Finds housing supply inelastic in such areas

Slope of over 15%: now often used as geographic constraints to housing supply / urban development

- Diamond (2016), Hariri (2015), Chen & Kung (2013)

4.3 IRREGULAR TERRAIN MODEL

Used by radio/tv enginneers to predict signal reception



(Figure 2 of Olken (2009))

APPLICATION 1: OLKEN (2009)

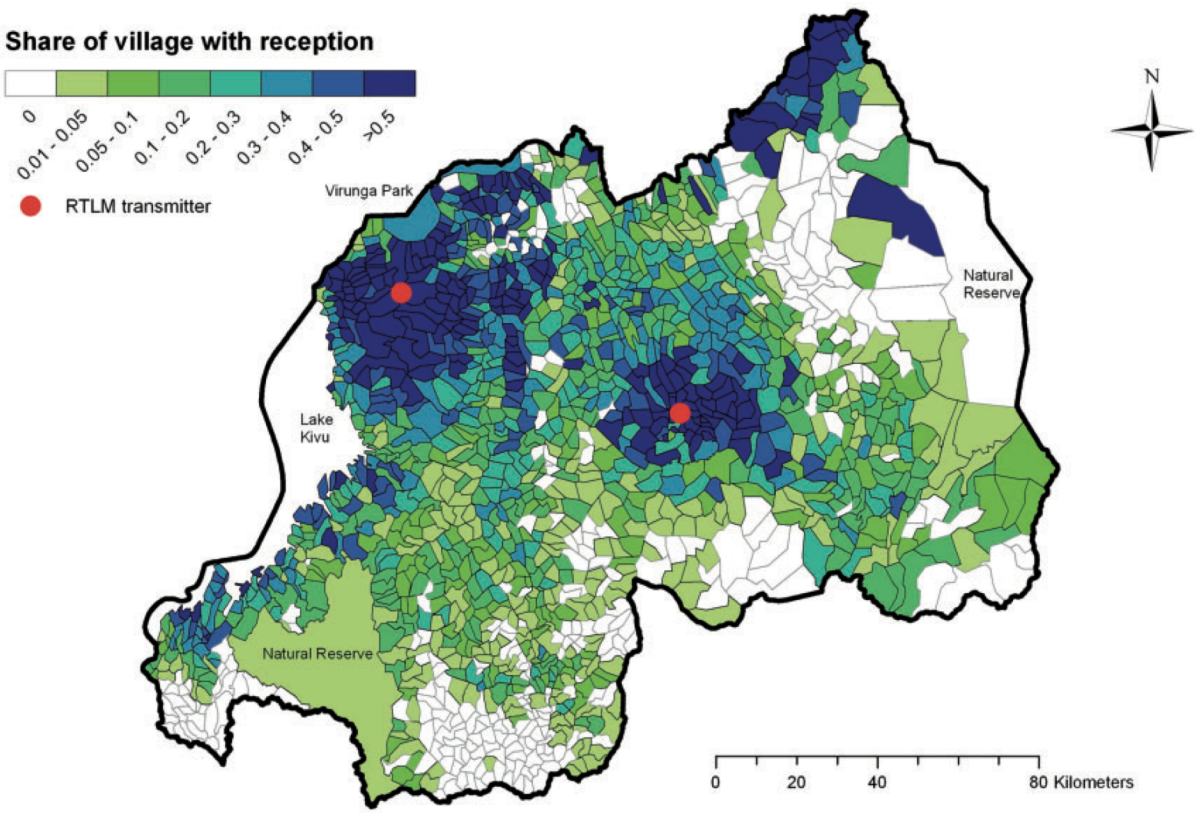
of TV channels ↑ in Indonesia ⇒ Social capital ↓

IV: **TV signal strength**

APPLICATION 2: YANAGIZAWA-DROTT (2014)

Anti-Hutu radio ⇒ Rwandan genocide incidents ↑

IV: **radio signal strength**



5. DISTANCE

Coordinate systems for distance

Buffer

Feature To Point + Near

Generate Near Table

5.1 COORDINATE SYSTEMS FOR DISTANCE

Two approaches:

1. Geographic coordinate systems

2. UTM projections

APPROACH 1: GEOGRAPHIC COORDINATE SYSTEMS

DISTANCE BETWEEN POINTS

1. Obtain geographic coordinates (How?)
2. Use the Great Circle Distance formula

$$d_{ij} = 111.12 \times \cos^{-1}\Big[\sin(La_i)\sin(La_j) + \cos(La_i)\cos(La_j)\cos(Lo_i - Lo_j)\Big]$$

   - d_{ij}: distance in km from i to j
   - Proof: see Wolfram MathWorld
   - Can be implemented by Stata ado `globdist`

APPROACH 1: GEOGRAPHIC COORDINATE SYSTEMS (CONT.)

DISTANCE FROM POLYGON

Obtain their centroids by:

- Stata ado `shp2dta`
- Feature To Point + Add XY Coordinates in ArcGIS

Then use `globdist`

APPROACH 1: GEOGRAPHIC COORDINATE SYSTEMS (CONT.)

DISTANCE TO POLYLINES

Obtain nearest point on polyline by:

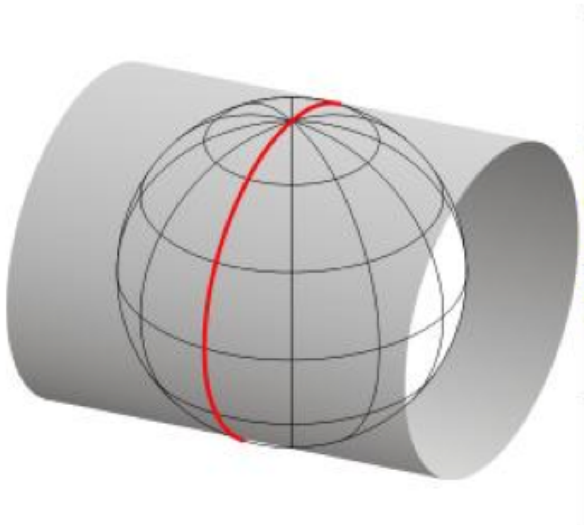
- Near tool in ArcGIS

Near tool calculates distance as well

APPROACH 2: UTM PROJECTIONS

(UTM = Universal Transverse Mercator)

Project earth surface onto the cylinder that is tangent on **standard meridian**

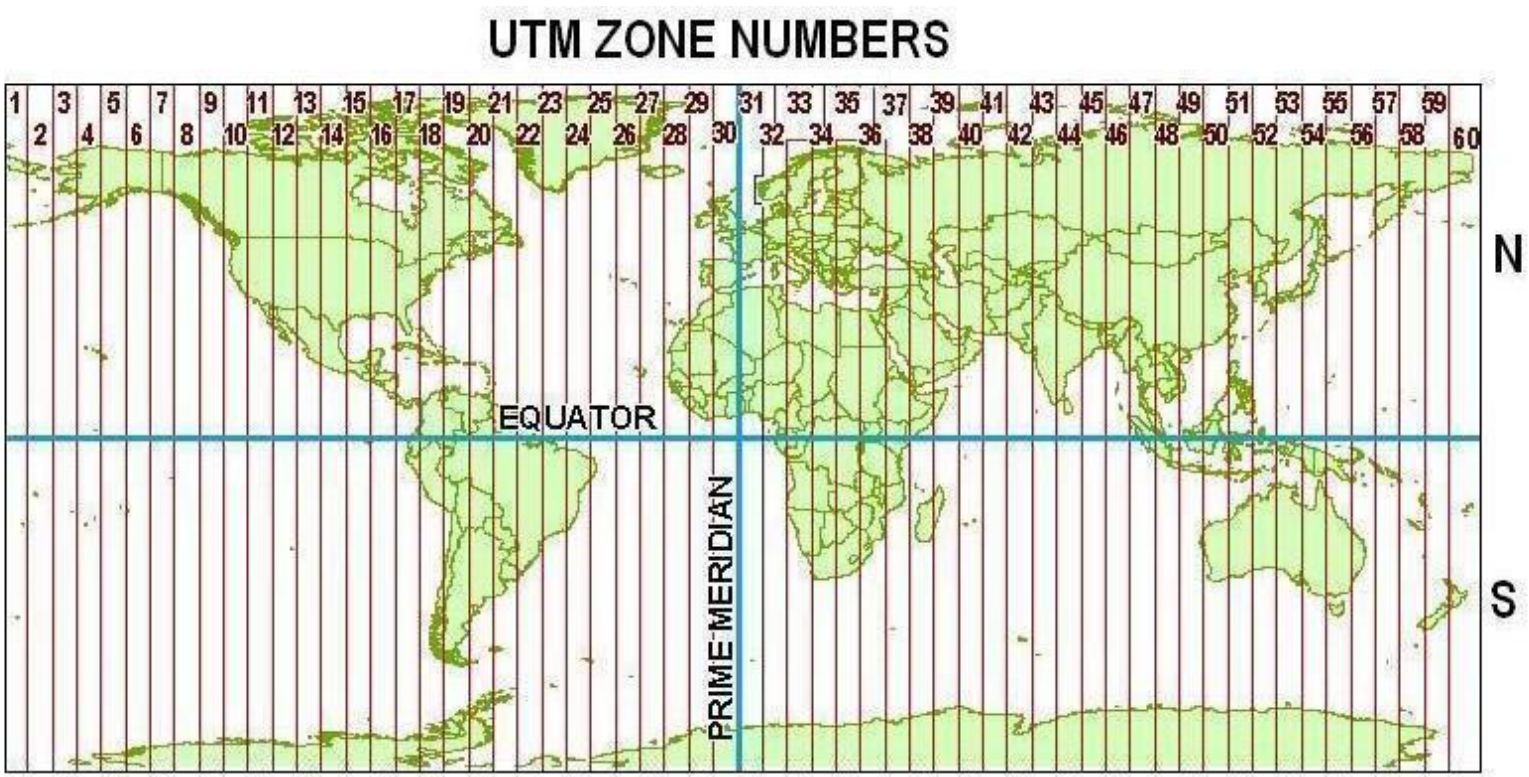


Father away from standard meridian, more distortion

APPROACH 2: UTM PROJECTIONS (CONT.)

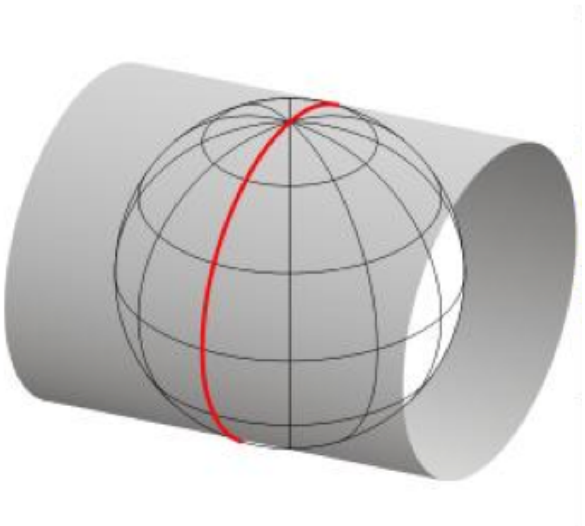
To minimize distortion:

1. Divide Earth into 60 zones (6° wide in longitude)



APPROACH 2: UTM PROJECTIONS (CONT.)

2. For each zone, set standard meridian in the middle

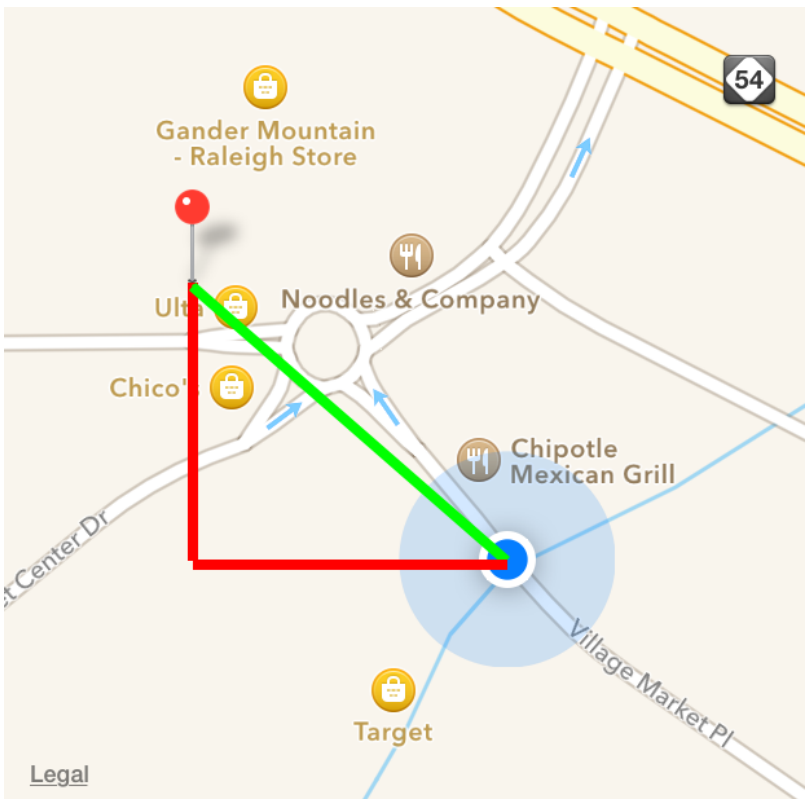


APPROACH 2: UTM PROJECTIONS (CONT.)

3. Scale down distance along standard meridian by 0.9996
   - This number is called **scale factor**
   - 0.9996 minimizes overall distortion w/i the 6°-wide zone

APPROACH 2: UTM PROJECTIONS (CONT.)

Once projected, use Pythagoras' Theorem



DISTANCE CALCULATION IN ARCGIS

Tools such as Buffer and Near calculate distance

GEODESIC option for geographic coordinates

PLANAR option for UTM projections

LENGTH OF POLYLINES

e.g. River lengths, road distances

Geographic coordinates cannot be used

⇒ Use UTM projections

If the study area is large, cut polylines by UTM zones

LENGTH OF POLYLINES (CONT.)

ARCGIS TOOLS TO USE FOR 2D LENGTH:

Project

Add Field + Calculate Field with expression:

```
float(!shape.length!)
```

LENGTH OF POLYLINES (CONT.)

ARCGIS TOOLS TO USE FOR 3D LENGTH:

Project (for polylines)

Project Raster (for elevation raster)

Add Surface Information, with SURFACE_LENGTH option

APPLICATIONS

Duflo & Pande (2007): length of rivers by district as control

Dell (2010): length of roads by district as intermediate outcome

SHORTEST PATH PROBLEM

Use ArcGIS tools such as:

- Cost Distance
- Cost Path
- Path Distance
- Corridor

See Melissa Dell's lecture note (pp. 18-25) for detail

5.2 BUFFER

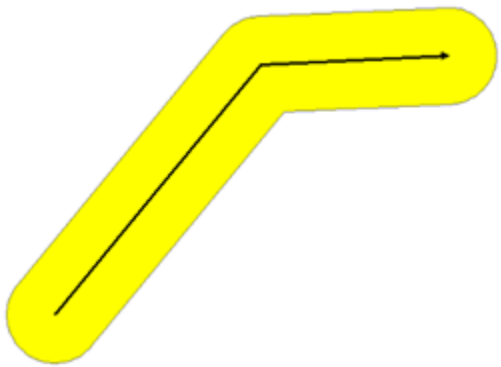
Creates neighborhood polygons

Size of neighborhood can be specified

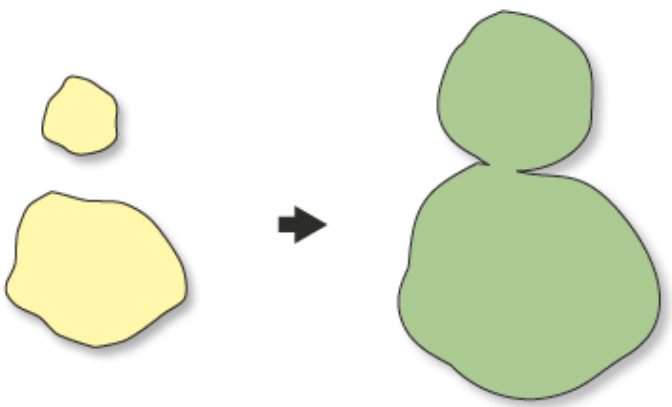
For points: a circle of x-meter/feet radius

5.2 BUFFER (CONT.)

For polylines:



For polygons:



5.2 BUFFER (CONT.)

One useful application:

Identify other features in the neighborhood

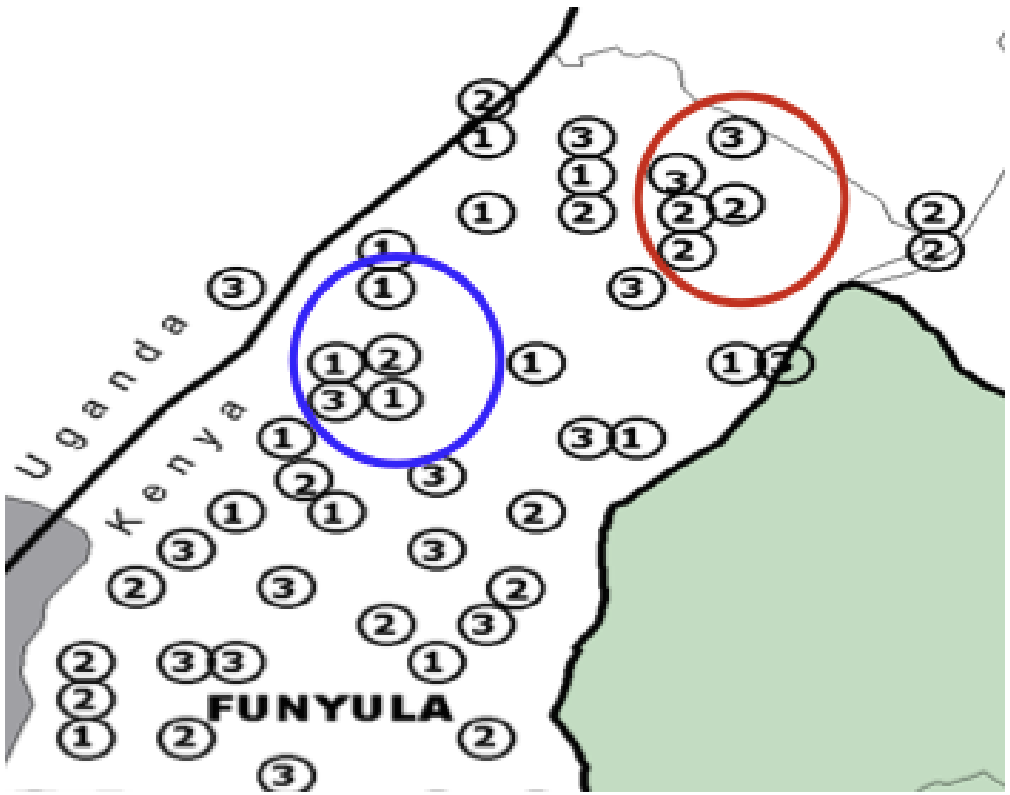
⇐ Use buffer polygons as target features in Spatial Join with JOIN_ONE_TO_MANY options

PYTHON CODE FOR IDENTIFYING NEIGHBORS

```
import arcpy

# Identify other villages within 20-mile radius

in_features = "villages.shp"
radius = "20 miles"

buffer = "village_buffer.shp"
out_feature_class = "village_neighbors.shp"

arcpy.Buffer_analysis(in_features, buffer, radius)
arcpy.SpatialJoin_analysis(
        buffer, in_features,
        out_feature_class, "JOIN_ONE_TO_MANY")
```

APPLICATION 1: MIGUEL & KREMER (2004)

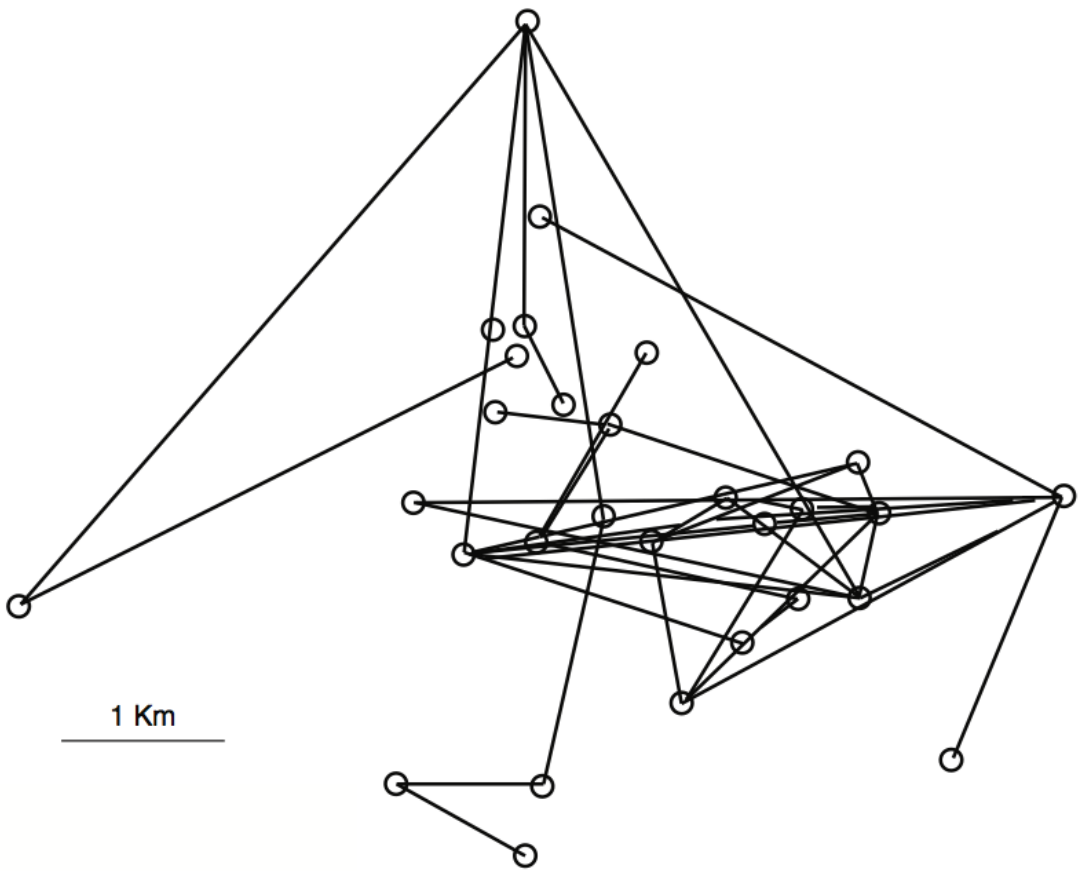
Deworming school children ⇒ Infection in neighborhood ↓?



(Image source)

APPLICATION 2: CONLEY & UDRY (2010)

Farmer's use of new technology ⇒ Learning by friends?



Control for average use in neighborhood (1km radius)

APPLICATION 3: MUEHLENBACHS ET AL. (2015)

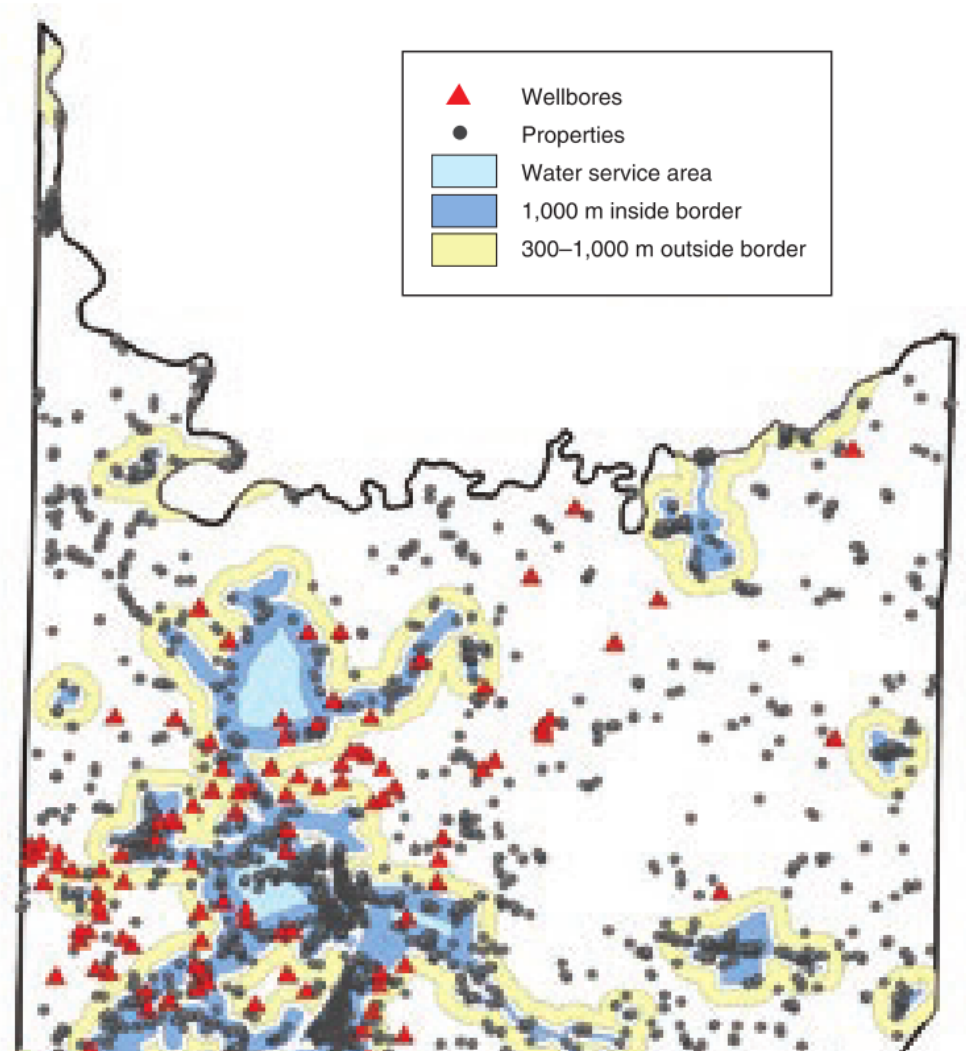
Shale gas $\Rightarrow$ House price $\downarrow$?

Treatment: # of drilled wells w/i 2km radius of each house

APPLICATION 3: MUEHLENBACHS ET AL. (2015) (CONT.)

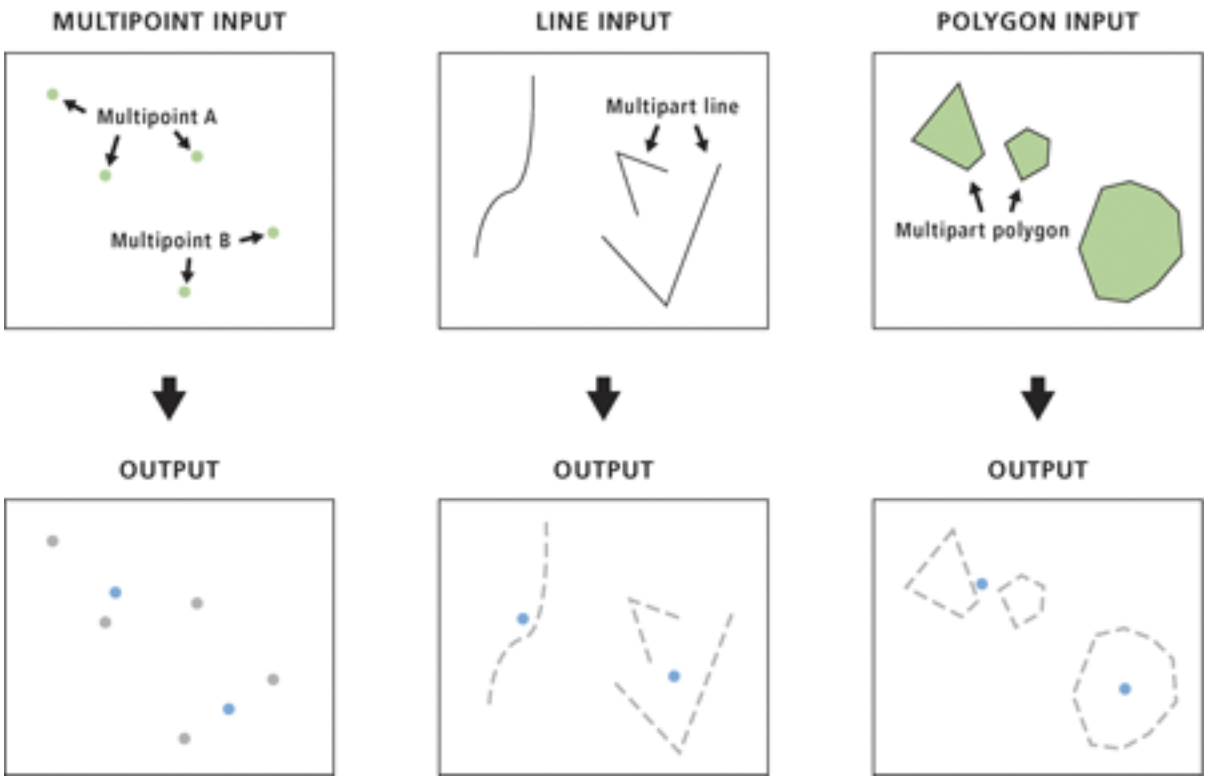
Sample restriction: 1km buffer along water service area boundary

(Figure 6 of Muelenbachs et al. 2015)



5.3 FEATURE TO POINT + NEAR

Feature To Point : Creates centroids of input features



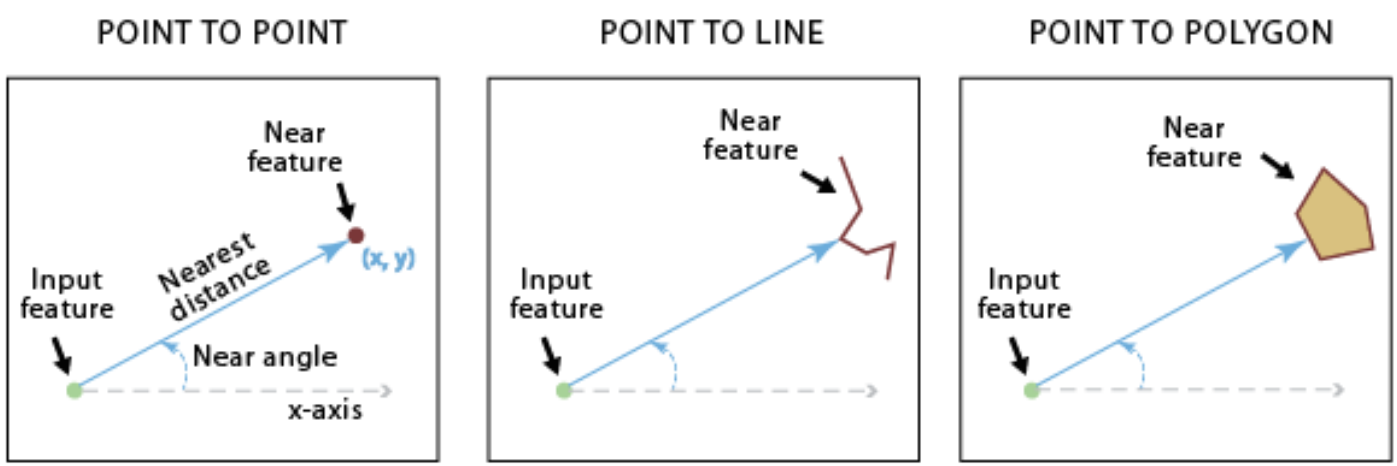
5.3 FEATURE TO POINT + NEAR (CONT.)

Near: Finds nearest feature to each input feature

Also calculates distance to it

If nearest features are polyline / polygon

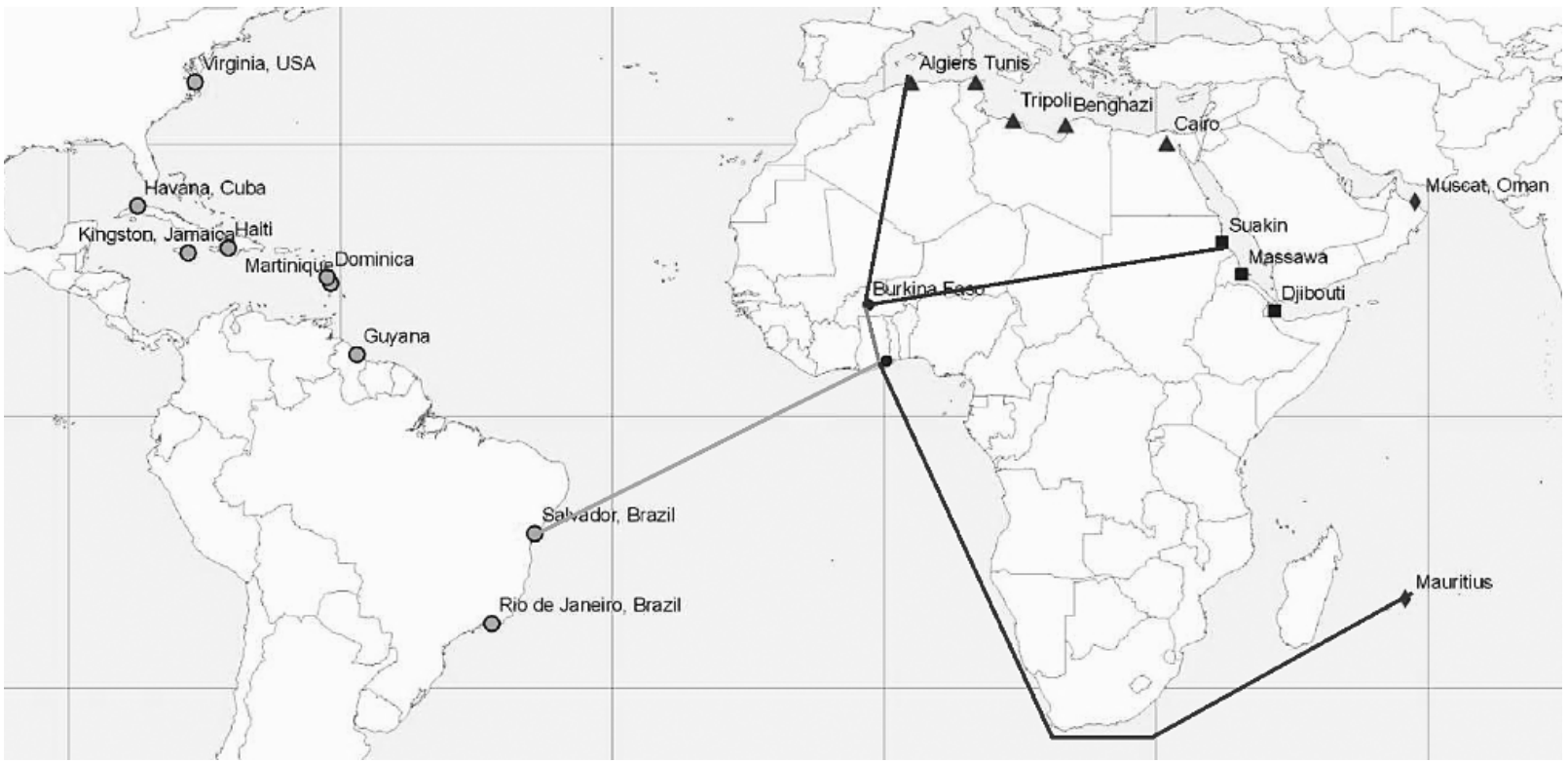
⇒ Can obtain coordinate of nearest point



APPLICATION 1: NUNN (2008)

Slave trade ⇒ Africa's underdevelopment?

IV for slave trade: Distance to nearest slave trade centers



APPLICATION 1: NUNN (2008) (CONT.)

Feature To Point ⇒ Country centroids

Near w/ coastline ⇒ Coordinates of nearest coastal point

Make XY Event Layer + Copy Feature ⇒ Nearest coastal point

Near w/ slave trade centers ⇒ Distance to nearest center

APPLICATION 1: NUNN (2008) (CONT.)

EXCLUSION RESTRICTIONS

Location of slave trade centers:

- Determined by climate suitability of plantation crops / location of mines (p. 160)
- Not affected by the distance to Africa

Distance to slave markets ≠ Distance to other economic opportunities

- Reduced-form correlation is absent outside Africa (p. 163)

APPLICATION 1: NUNN (2008) (CONT.)

LATE

Maybe a smaller impact if countries voluntarily engaged in slave trades

But LATE may be of more interest in this context

APPLICATION 2: ROGALL (2014)

In Rwandan genocide:

Presence of armed groups ⇒ civilian participation ↑?

Instrument:

Distance to main road

x

Rainfall along the dirt track to main road during genocide

APPLICATION 2: ROGALL (2014) (CONT.)

Distance to main road

- Feature To Point
- Near

Rainfall along the dirt track to main road during genocide

- XY To Line
- Buffer
- Spatial Join

5.4 GENERATE NEAR TABLE

Calculates distance to many features, besides nearest one

APPLICATION 1: CAMPANTE & DO (2014)

Population concentration around capital

$\Rightarrow$ US state govt quality $\uparrow$

- Newspaper coverage of state politics $\uparrow$
- Money politics $\downarrow$
- Public good provision $\uparrow$

APPLICATION 1: CAMPANTE & DO (2014) (CONT.)

Measure population concentration around capital by:

- Feature To Point ⇒ County centroids
- Generate Near Table ⇒ Distance from capital to county centroids
- In Stata, obtain the weighted sum of population w/ inverse of distance as weight

APPLICATION 2: CURRIE & NEIDELL (2005)

Air pollution ⇒ Infant mortality in California?

Data:

- Infant mortality: individual-level with zip-code of mothers' residency
- Air pollution: monitor-level (geo-referenced)

MONITOR LOCATIONS & ZIP-CODE POLYGONS

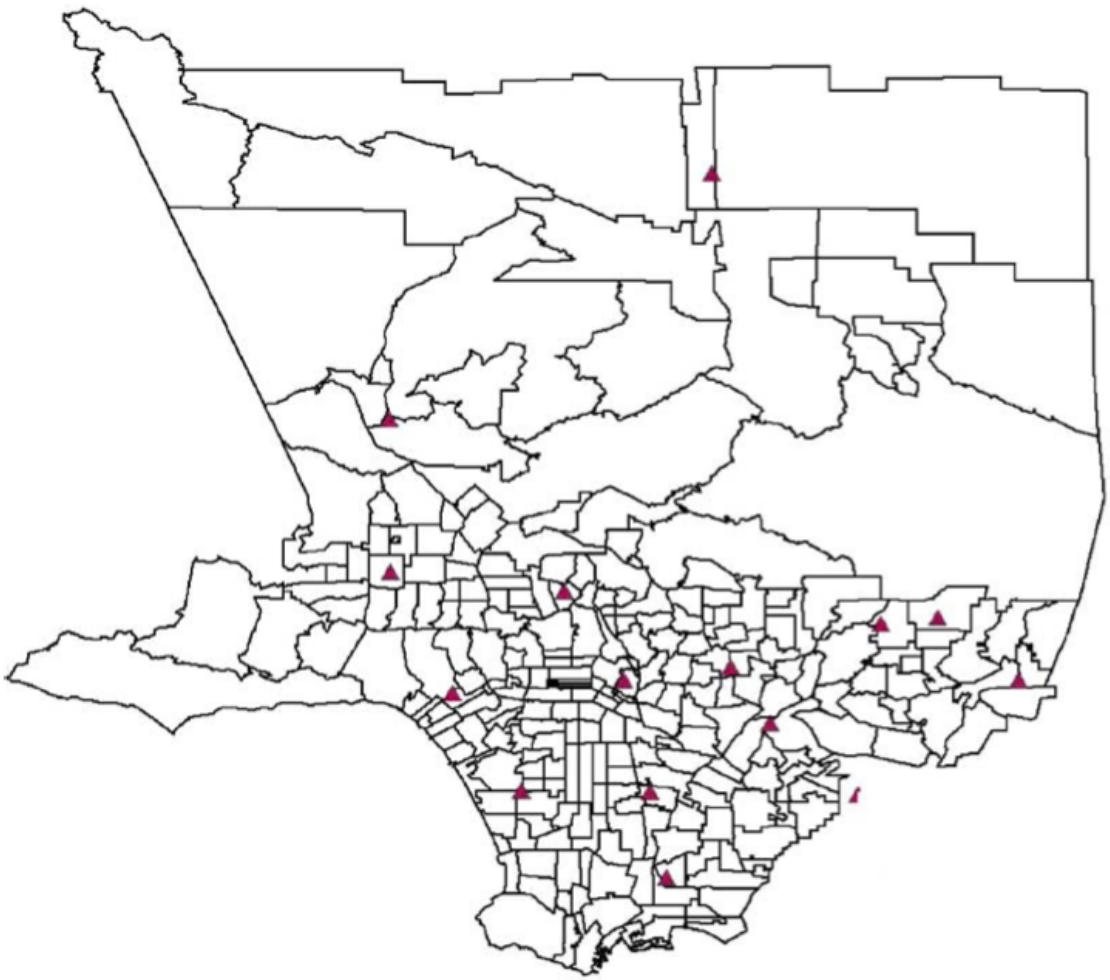


Fig. 4. Ozone monitors in Los Angeles county.

(taken from Neidell 2004)

APPLICATION 2: CURRIE & NEIDELL (2005) (CONT.)

How to construct pollution measure at zip-code level?

Feature To Point

⇒ Zip-code zone centroids

Generate Near Table

⇒ Distance from zip-code centroid to each monitor

Average pollution by weighting each monitor with inverse distance (within 20 miles)

6. SPATIAL REGRESSIOIN DISCONTINUITY

WHAT'S UNIQUE ABOUT SPATIAL RD

Forcing variable: two-dimensional vector (i.e. coordinates)

Cutoff: boundary lines

$\Rightarrow$ How should we extend the standard RD design?

APPROACH 1: SCALAR RD

$$y_i = \beta T_i + \gamma Dist_i + \delta T_i Dist_i + \mu_s + \varepsilon_i$$

Project coordinates into distance to boundary ($Dist_i$)

- $Dist_i < 0$ for control areas

APPROACH 1: SCALAR RD (CONT.)

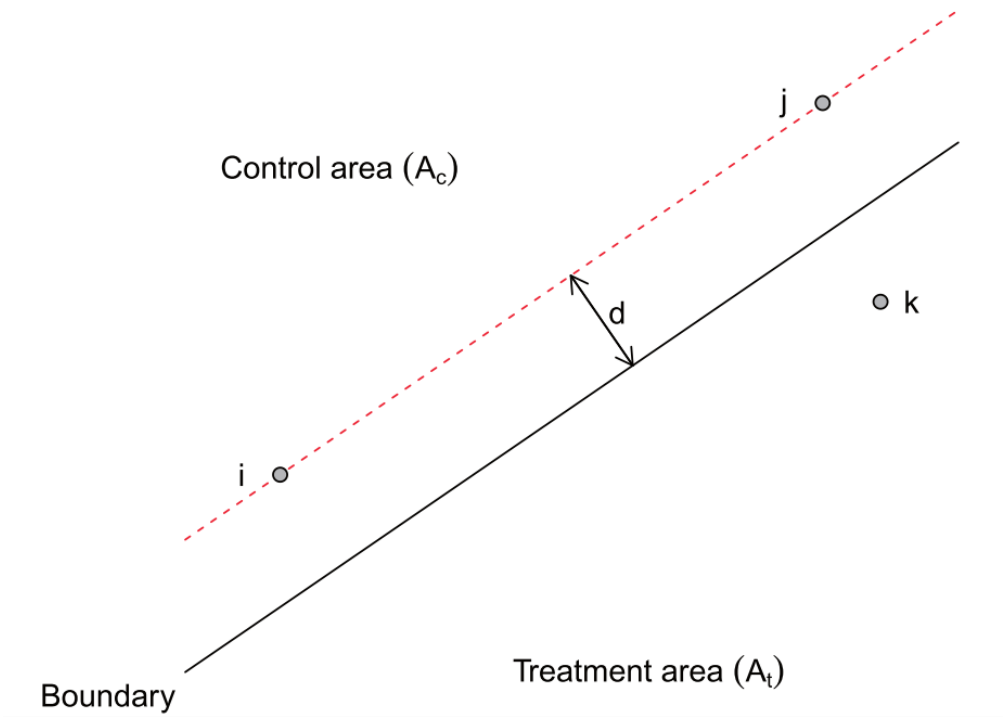
$$y_i = \beta T_i + \gamma Dist_i + \delta T_i Dist_i + \mu_s + \varepsilon_i$$

Then use standard RD design w/ boundary segment FE (μ_s)

- i's nearest point on boundary $\in$ Segment s

APPROACH 1: SCALAR RD (CONT.)

w/o boundary segment FE, you compare point i to k in diagram below



(Figure 3 of Keele & Titiunik 2014)

APPROACH 2: BOUNDARY RD

$$y_i = \beta T_i + \boldsymbol{x}_i'\boldsymbol{\gamma} + T_i\boldsymbol{x}_i'\boldsymbol{\delta} + \mu_n + \varepsilon_i$$

Use coordinates ($\boldsymbol{x}_i$) for RD polynomials

- Not zero at boundary

APPROACH 2: BOUNDARY RD (CONT.)

$$y_i = \beta T_i + \boldsymbol{x}_i'\boldsymbol{\gamma} + T_i\boldsymbol{x}_i'\boldsymbol{\delta} + \mu_n + \varepsilon_i$$

Pick (equally-spaced) points on boundary (denoted by n)

Assign each observation to its nearest boundary point, to define μ_n

APPROACH 2: BOUNDARY RD (CONT.)

$$y_i = \beta T_i + \boldsymbol{x}_i' \boldsymbol{\gamma} + T_i \boldsymbol{x}_i' \boldsymbol{\delta} + \mu_n + \varepsilon_i$$

Treatment effect at boundary point n: given by

$$\beta + \boldsymbol{x}_n' \boldsymbol{\delta}$$

See Chapter 2 of Zajonc (2012) (commonly cited as Imbens and Zajonc 2011) for more detail

DATA GENERATION FOR SPATIAL RD

1. Attach treatment indicator to zone polygons
2. Treatment boundary polylines
3. Boundary segment indicator & Distance to boundary
4. Visualize spatial RD plots

1. ATTACH TREATMENT INDICATOR

Create a text file table of:

- Polygon ID
- Treatment indicator

Table To Table to convert it into **dBASE format**

- Format for attribute tables in ArcGIS

Join Field to merge the data by polygon ID

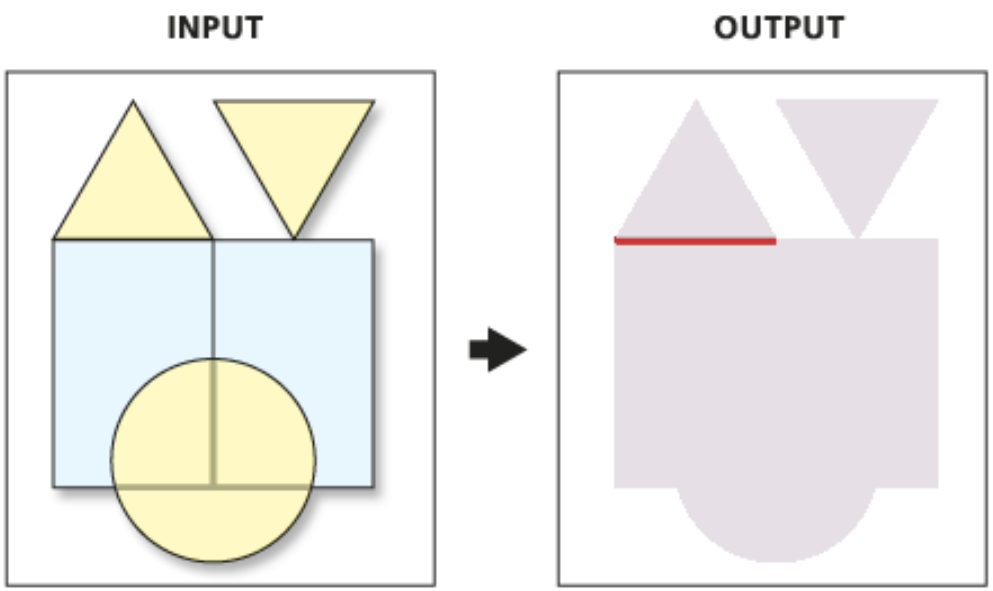
- It's ArcGIS's version of Stata's `merge`

2. TREATMENT BOUNDARY

Select to create two polygon shapefiles

- One for treated zones
- The other for control zones

Intersect with output type LINE



3. BOUNDARY SEGMENT INDICATOR

First, split boundary polyline into segments of equal length

- **Unsplit Line**
- **Add Field** + **Calculate Field** with expression:

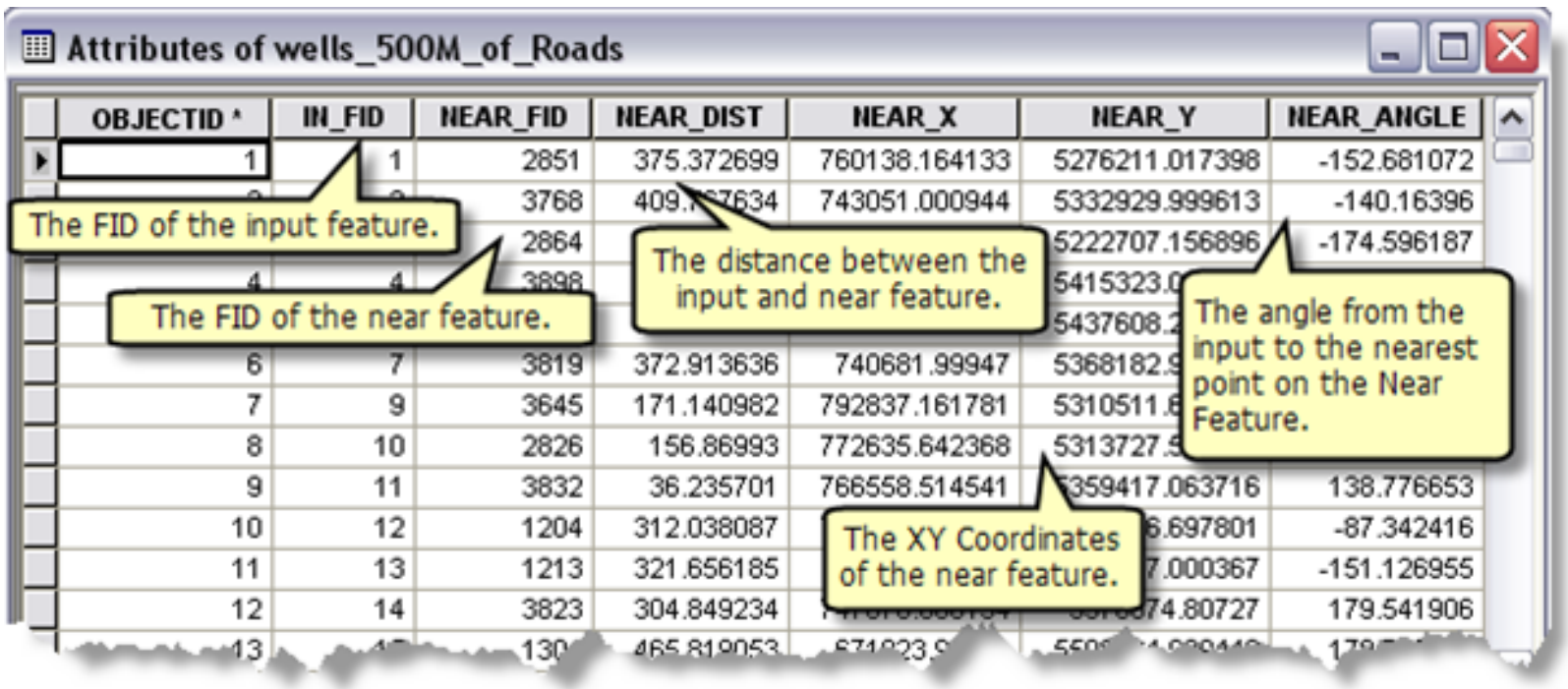
```
!shape!.positionAlongLine(0.5,True).firstPoint.X
```

- **Make XY Event Layer** + **Copy Features** ⇒ Split points
- **Split Line At Point**

3. BOUNDARY SEGMENT INDICATOR (CONT.)

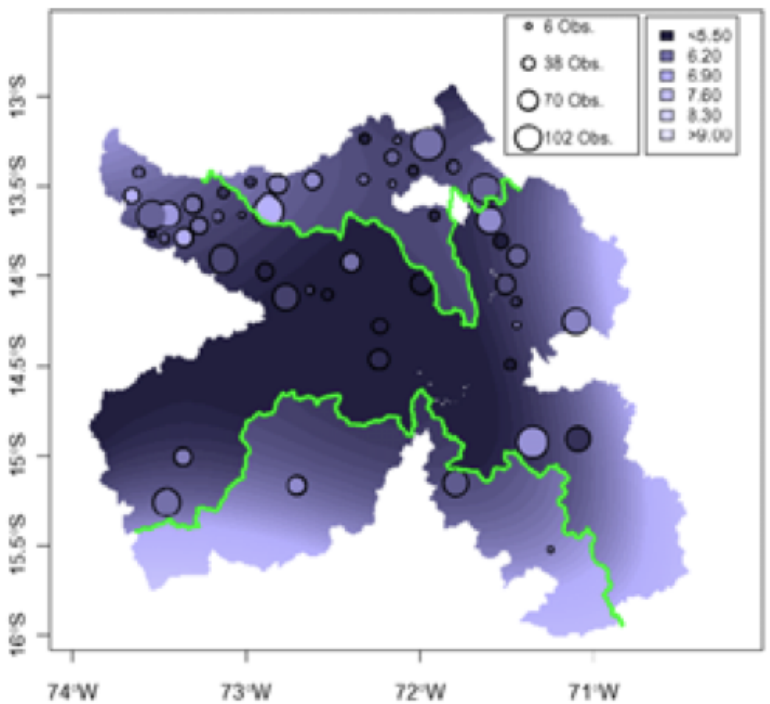
Then, use **Near**

- Nearest segment feature ID (NEAR_FID)
- Distance to boundary (NEAR_DIST)

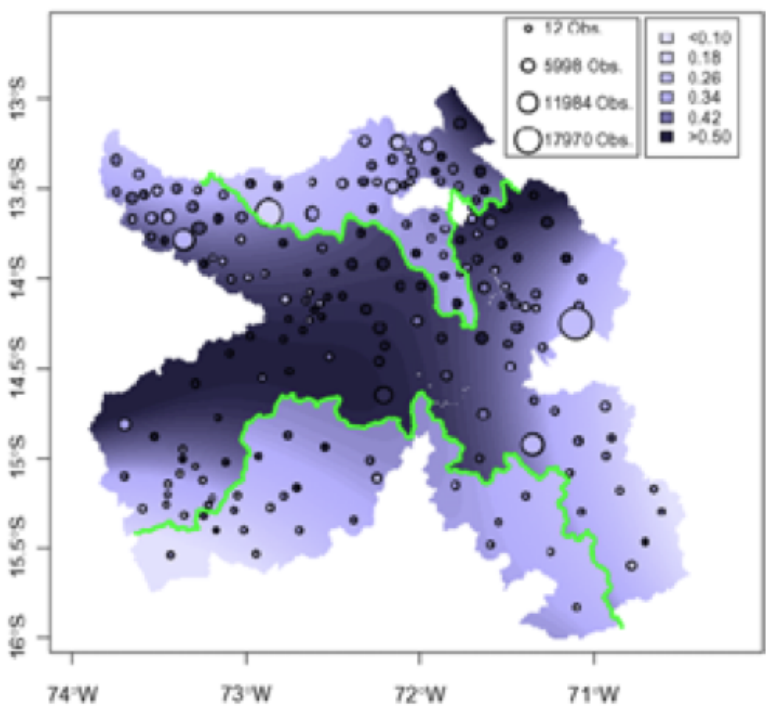


APPLICATION 1: DELL (2010)

Does forced labor system during Spanish colonial rule affect todya's living standards in Peru? If so, why?



(a) Consumption (2001)



(b) Stunting (2005)

APPLICATION 2: MICHALOPOULOS & PAPAIOANNOU (2014)

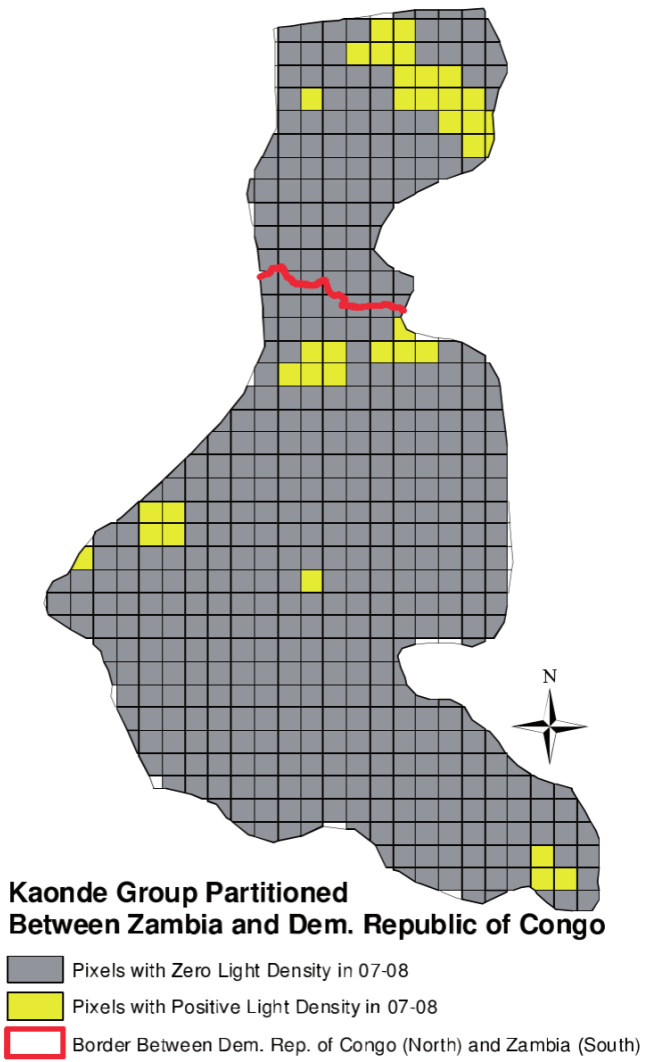
Ntnl govt quality ⇒ development?

Look at ethnic homelands split by national boarders in Africa

- Segment FE = Ethnicity FE

Find no difference in nighttime light across border on average

Large heterogeneity across ethnic groups, though



BERGER ET AL. (2016)

Does higher TV license fees increase evasion in Austria?

Exploit differences in fees across states

Focus on state borders where covariates are balanced

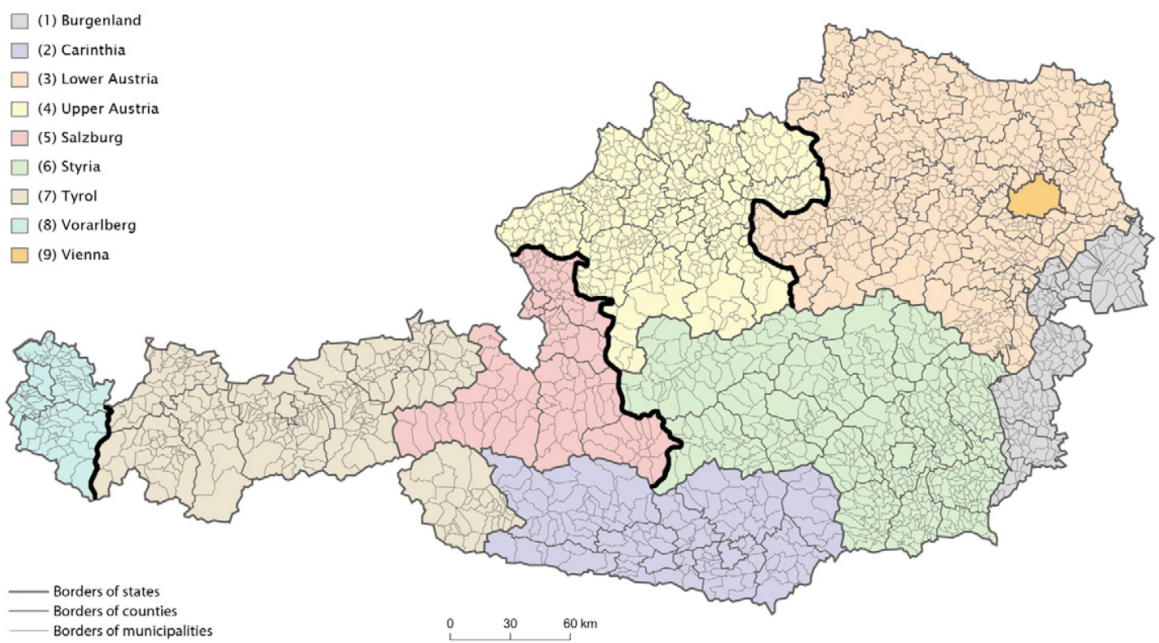


Fig. 1. Austrian state borders. *Notes:* The state borders in bold indicate the 'most balanced' borders.

GONZALEZ (2015)

Cellphone coverage ⇒ Electoral fraud ↓ in Afghanistan

1000+ polling stations w/i 5km from coverage boundary

⇒ Boundary RD approach: feasible

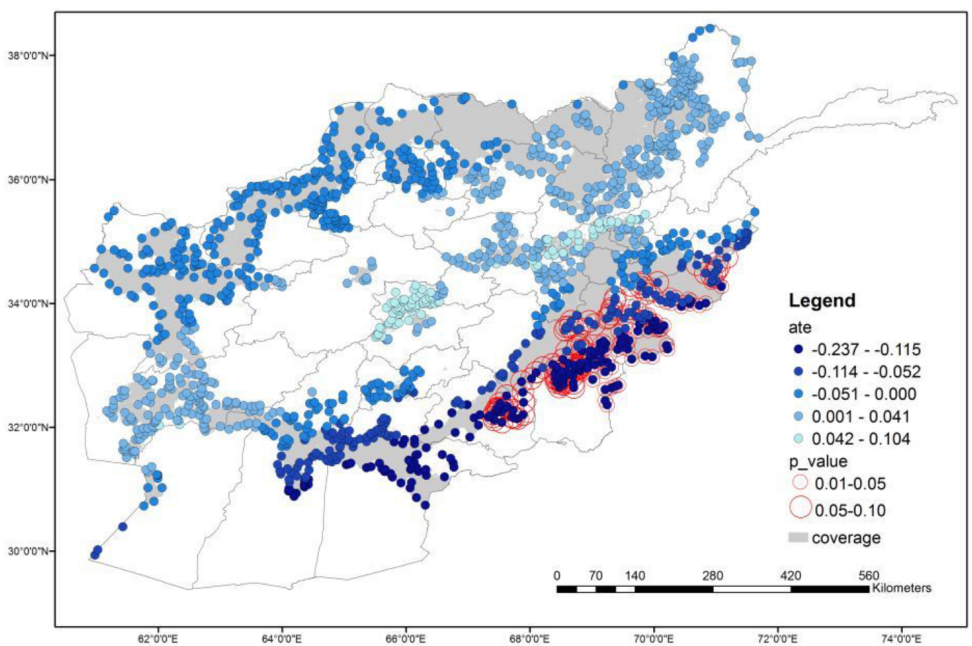


Figure 6: Spatial Distribution of Boundary Treatment Effects (Category C+ fraud)

7. SURFACE AREA

COORDINATE SYSTEMS FOR SURFACE AREA

Geographic coordinate systems: not suitable

- $1°$ in lat = 110.6 km at equator / 111.7 km at poles
- $1°$ in lon = 111.3 km at equator / 55.8 km at 60° N/S

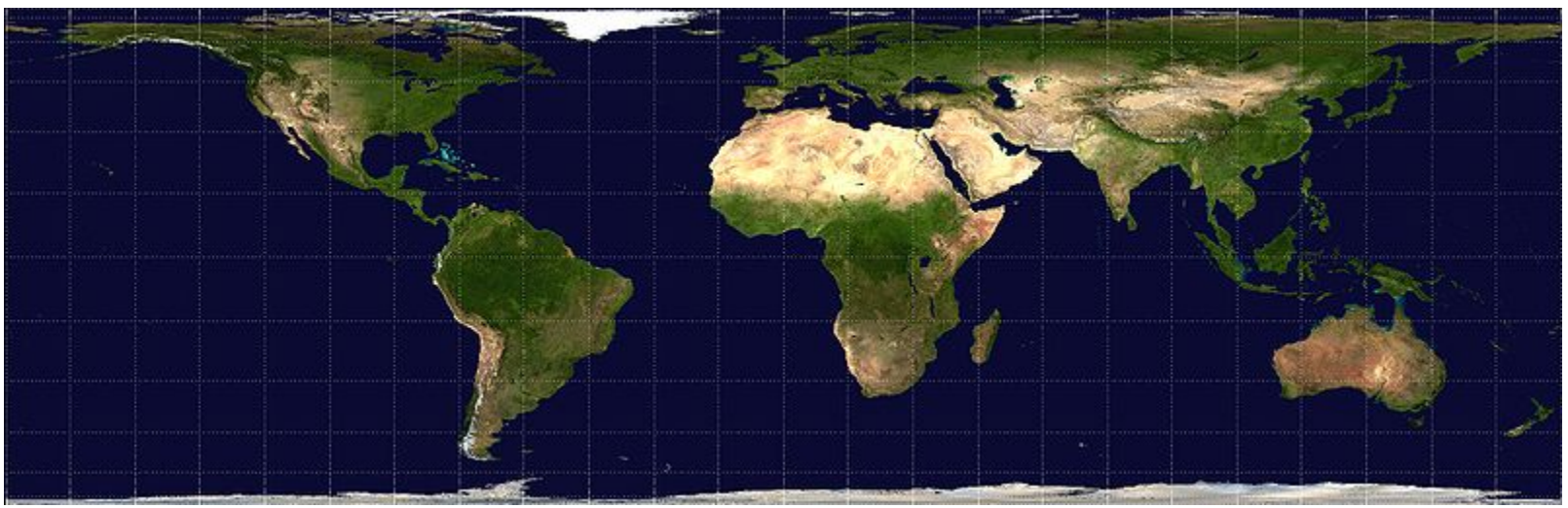
⇒ Use either:

- UTM projections (for small areas)
- Equal Area projections (for large areas)

EQUAL AREA PROJECTION #1

LAMBERT CYLINDRICAL EQUAL AREA

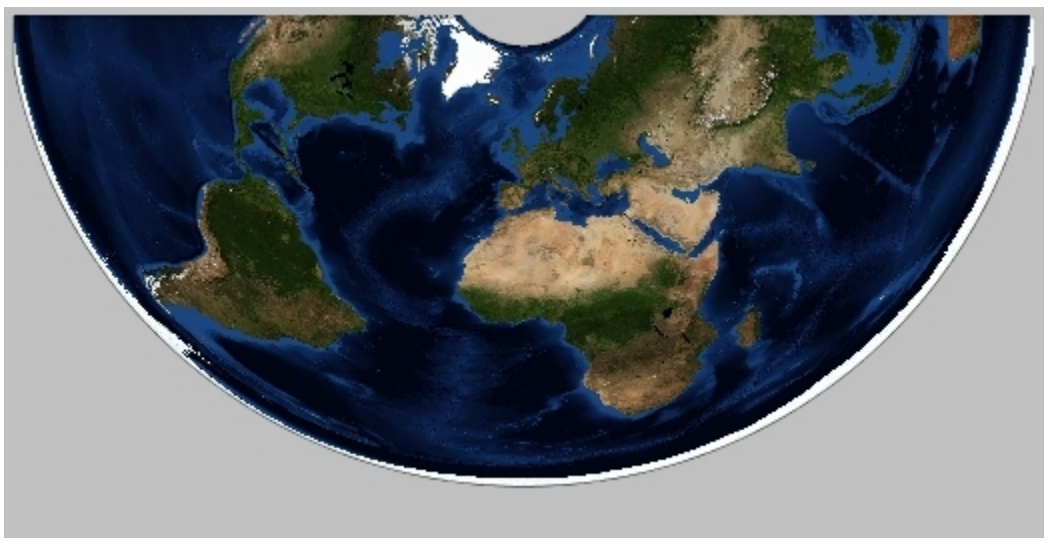
Shrink latitude to compensate shorter unit of longitude towards Poles



EQUAL AREA PROJECTION #2

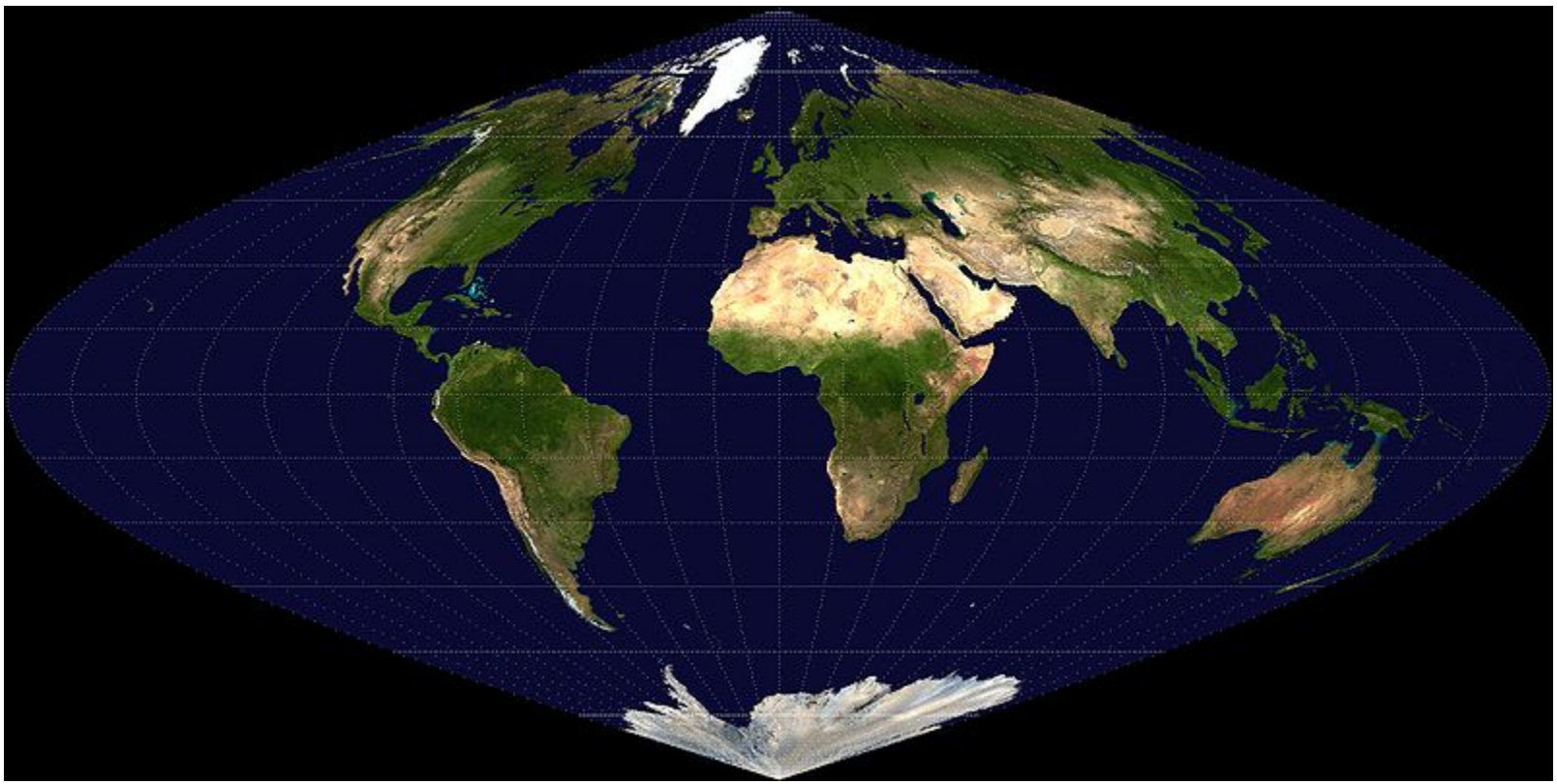
ALBERTS CONIC EQUAL AREA

Standard map projection for USA

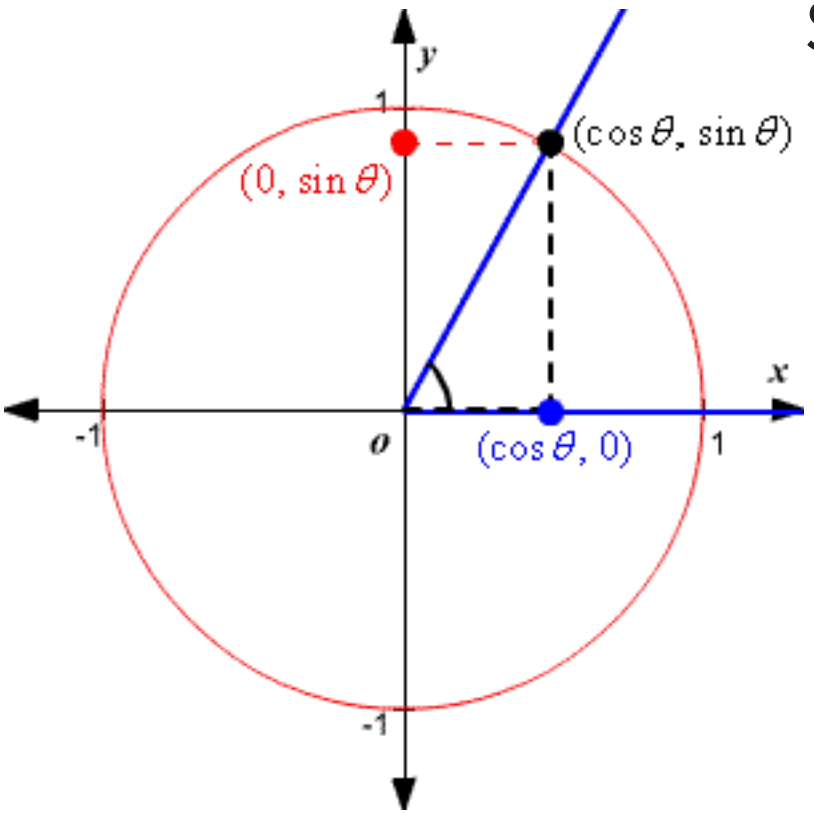


EQUAL AREA PROJECTION #3

SINUSOIDAL PROJECTION



SINUSOIDAL PROJECTION (CONT.)



Shrink longitude by $\cos(latitude)$

EQUAL AREA PROJECTIONS

All give the (approximately) correct surface area

Difference is in how the projected map looks

CALCULATE SURFACE AREA IN ARCGIS

Project

Add Field + **Calculate Field** with expression:

```
float(!SHAPE.AREA!)
```

APPLICATION: NUNN (2008)

Assign # of slaves from each ethnic group into different countries by surface area

Figure II of Nunn (2008)

8. MAP ALGEBRA

Cell-by-cell calculation across multiple raster datasets

1. Arithmetic operations
   - with numbers (e.g., `raster*2`)
   - across several raster datasets (e.g., `ras1 + ras2`)
2. Functions
   - Math toolset (square root, logarithm, sine/cosine, etc.)
   - Cell Statistics
   - Focal Statistics

APPLICATION 1: MAYSHAR ET AL. (2015)

What caused the formation of the state?

Focus on crop type (cereals vs roots/tubers)

THEORY

ROOTS AND TUBERS

Cassava, yam, taro, bananas...

Perishable upon harvest

Harvesting is non-seasonal

⇒ No incentive to steal / confiscate

THEORY (CONT.)

CEREALS

Wheat, rice, maize...

Storable

Harvest within a short period of time

⇒ Incentive to steal / confiscate

THEORY (CONT.)

STATE FORMATION

Building a state incurs a fixed cost

Bandits have no incentive to build a state with roots & tubers

Historical examples

- Ancinet Egypt: wheat with state
- New Guinea: yam/taro without state

DATA ON CEREAL / TUBER PRODUCTIVITY

GAEZ

Resolution: 5 x 5 arc-minutes (about 10 X 10 km)

Data: potential yields based on climate & soil

⇒ Exogeneous to human activities

⇒ Widely used by economists

- Nunn & Qian (2011), Costinot et al. (2016), etc.

MAP ALGEBRA IN ACTION

1. Convert yields into calorie for 15 crops

```
# Setting up
import arcpy
arcpy.CheckOutExtension("spatial")

# Input raster
maize_yield = arcpy.sa.Raster("maize.tif")

# Map algebra
maize_calorie = maize_yield * 36.5

# Save output
maize_calorie.save("calorie_cereal_maize.tif")
```

MAP ALGEBRA IN ACTION (CONT.)

2. Obtain maximum for each crop type

```
# Maximum calorie by cereal crops
input_cereals = arcpy.ListRasters(
        "calorie_cereal_*", "TIF")
max_cereal = CellStatistics(
        input_cereals, "MAXIMUM", "DATA")

# Maximum calorie by tuber/root crops
input_tubers = arcpy.ListRasters(
        "calorie_tuber_*", "TIF")
max_tuber = CellStatistics(
        input_tubers, "MAXIMUM", "DATA")
```

MAP ALGEBRA IN ACTION (CONT.)

2. Obtain maximum for each crop type

```
# Maximum calorie by cereal crops
input_cereals = arcpy.ListRasters(
        "calorie_cereal_*", "TIF")
max_cereal = CellStatistics(
        input_cereals, "MAXIMUM", "DATA")

# Maximum calorie by tuber/root crops
input_tubers = arcpy.ListRasters(
        "calorie_tuber_*", "TIF")
max_tuber = CellStatistics(
        input_tubers, "MAXIMUM", "DATA")
```

MAP ALGEBRA IN ACTION (CONT.)

3. Take difference

```
# Cereal's caloric advantage over tubers
caloric_diff = max_cereal - max_tuber

# Save the output
caloric_diff.save("caloric_diff.tif")
```

APPLICATION 2: NUNN & PUGA (2011)

Terrain ruggedness ⇒ income per capita

Negative impact outside Africa

- Transportation cost

Positive impact in Africa

- Negative once slave export is controlled for

TERRAIN RUGGEDNESS INDEX

Originally proposed by Riley et al. (1999)

Defined as:

$$TRI_{xy} = \sqrt{\sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} (e_{ij} - e_{xy})^2}$$

e_{xy} Elevation at longitude x latitude y

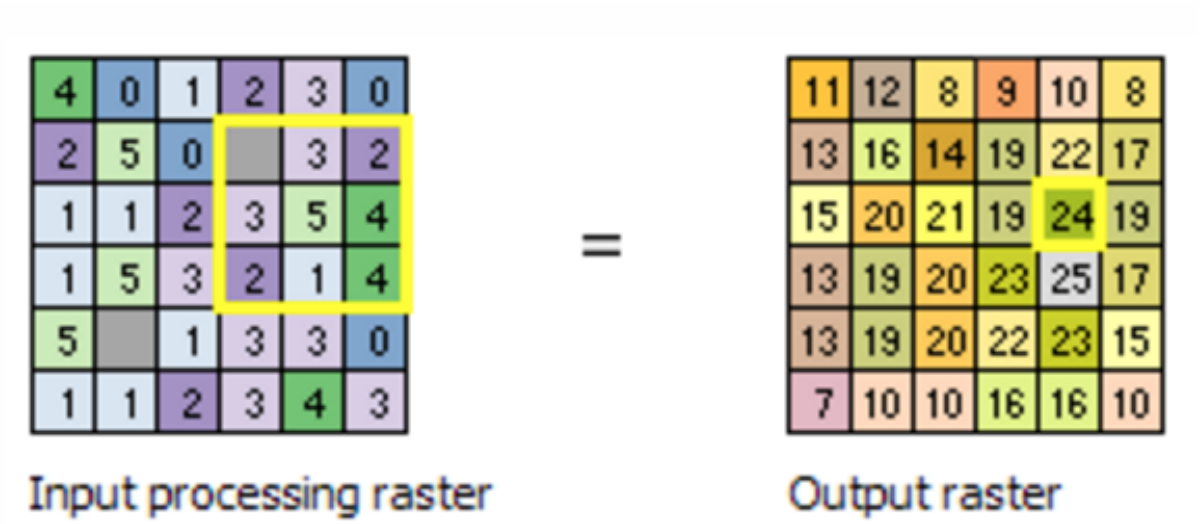
Can be obtained by **Focal Statistics** + Map Algebra

Raster cell (30x30 arc-sec) level data is downloadable from Diego Puga's website

FOCAL STATISTICS

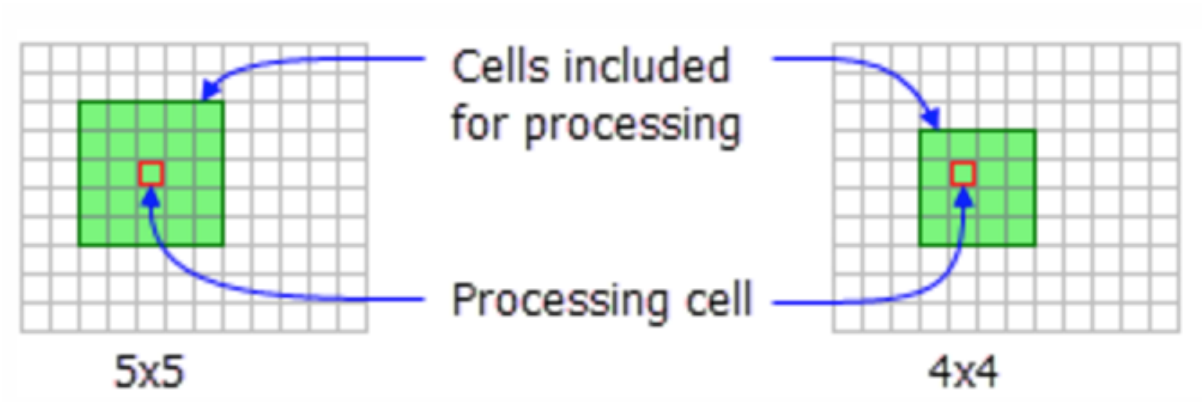
Calculates summary statistics in neighbouring raster cells

e.g. Sum of immediate neighbors



FOCAL STATISTICS (CONT.)

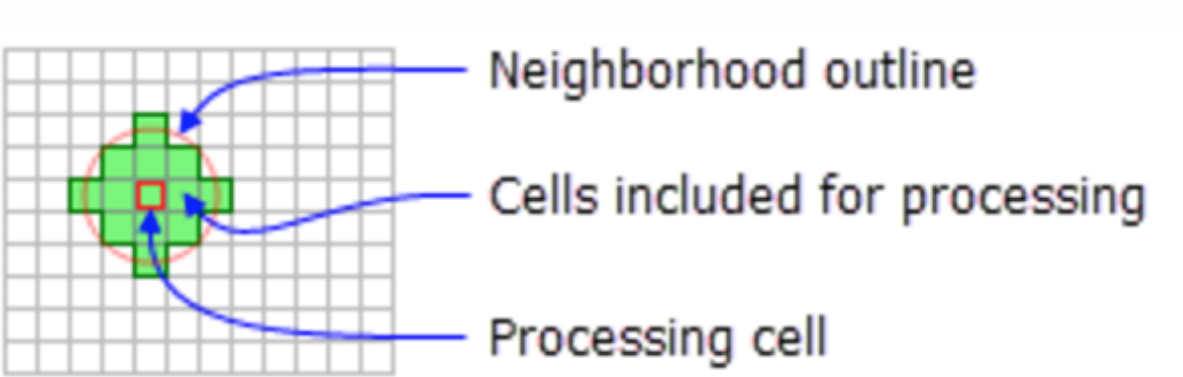
You can define the neighborhood very flexibly (more detail)



Rectangle

FOCAL STATISTICS (CONT.)

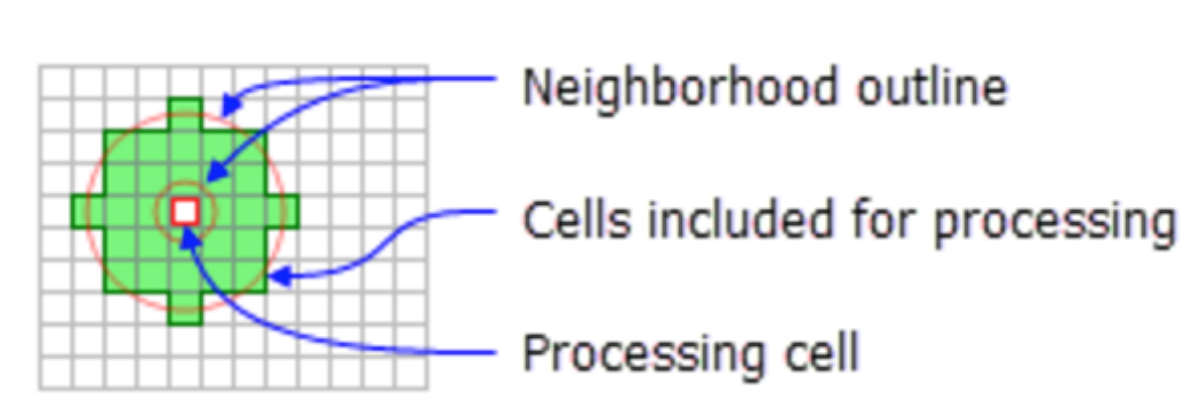
You can define the neighborhood very flexibly (more detail)



Circle

FOCAL STATISTICS (CONT.)

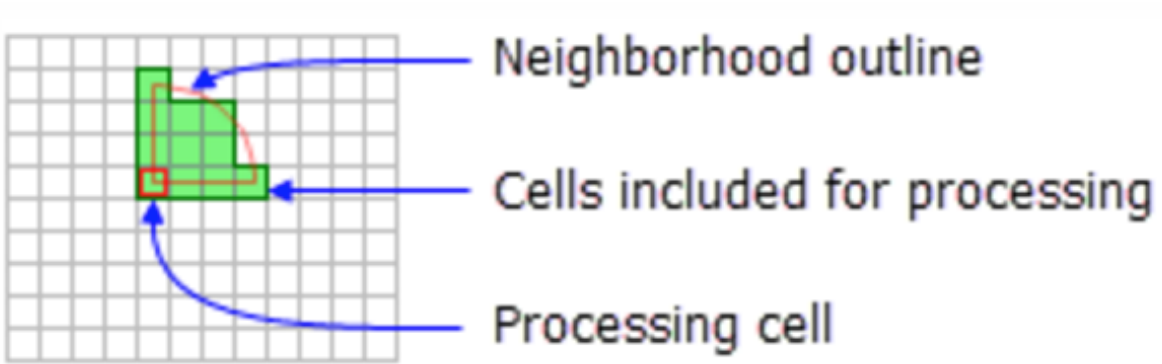
You can define the neighborhood very flexibly (more detail)



Annulus (ring)

FOCAL STATISTICS (CONT.)

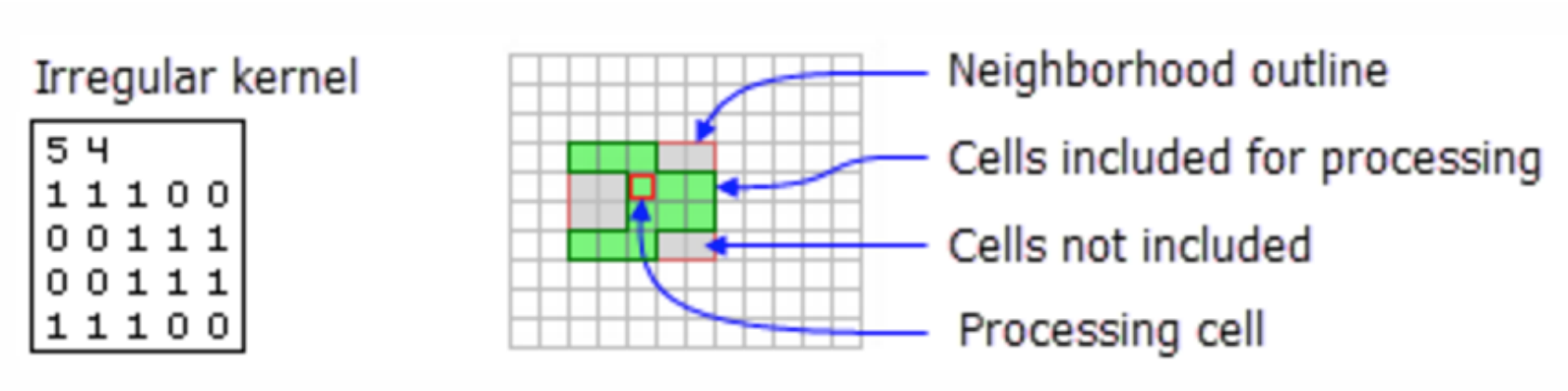
You can define the neighborhood very flexibly (more detail)



Wedge

FOCAL STATISTICS (CONT.)

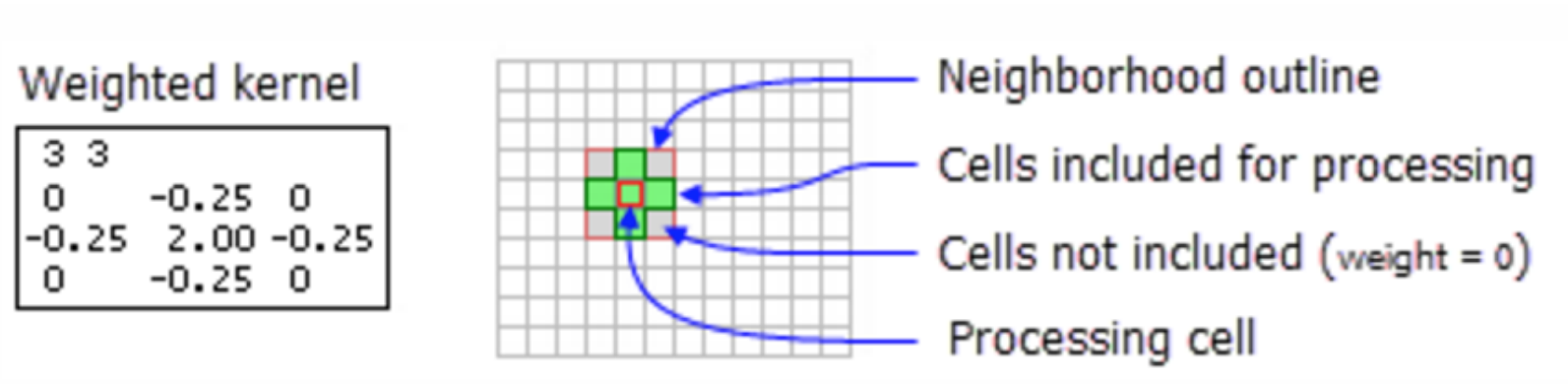
You can define the neighborhood very flexibly (more detail)



Irregular

FOCAL STATISTICS (CONT.)

You can define the neighborhood very flexibly (more detail)



Weight

TERRAIN RUGGEDNESS INDEX (CONT.)

Expand the expression inside the square root:

$$TRI_{xy} = \sqrt{\sum_i \sum_j (e_{ij} - e_{xy})^2}$$

$$= \sqrt{\sum_i \sum_j (e_{ij})^2 - 2e_{xy} \sum_i \sum_j e_{ij} + 9(e_{xy})^2}$$

TERRAIN RUGGEDNESS INDEX (CONT.)

$$TRI_{xy} = \sqrt{\sum_i \sum_j (e_{ij})^2 - 2e_{xy} \sum_i \sum_j e_{ij} + 9(e_{xy})^2}$$

Map Algebra calculates $(e_{ij})^2$:

```
elev = Raster("srtm30.tif")
elev_sq = elev**2
```

TERRAIN RUGGEDNESS INDEX (CONT.)

Expand the expression inside the square root:

$$TRI_{xy} = \sqrt{\sum_i \sum_j (e_{ij})^2 - 2e_{xy} \sum_i \sum_j e_{ij} + 9(e_{xy})^2}$$

Focal Statistics calculates $\sum_i \sum_j (e_{ij})^2$ and $\sum_i \sum_j e_{ij}$:

```
sum_elev_sq = FocalStatistics(elev_sq, "", "SUM", "")
sum_elev = FocalStatistics(elev, "", "SUM", "")
```

TERRAIN RUGGEDNESS INDEX (CONT.)

Expand the expression inside the square root:

$$TRI_{xy} = \sqrt{\sum_i \sum_j (e_{ij})^2 - 2e_{xy} \sum_i \sum_j e_{ij} + 9(e_{xy})^2}$$

Map Algebra sums them up and takes square root:

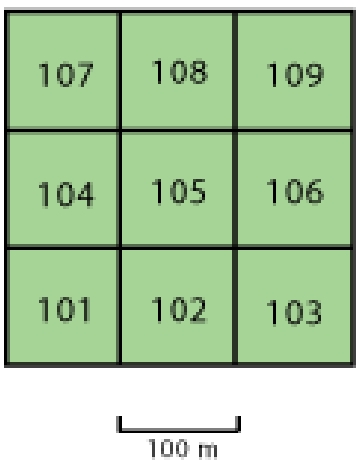
```
TRI_square = sum_elev_sq - 2*elev*sum_elev + 9*elev_sq
TRI = SquareRoot(TRI_square)
TRI.save("ruggedness.tif")
```

---

9. OTHER GEO-PROCESSING TOOLS USED BY ECONOMISTS

POLYGON NEIGHBORS

INPUT POLYGONS



OUTPUT TABLE

OBJECTID *	src_myCode	nbr_myCode	LENGTH	NODE _COUNT
1	101	102	100	0
2	101	104	100	0
3	101	105	0	1
4	102	101	100	0
5	102	103	100	0
6	102	104	0	1
7	102	105	100	0
8	102	106	0	1
9	103	102	100	0
10	103	105	0	1
11	103	106	100	0
12	104	101	100	0

APPLICATION OF POLYGON NEIGHBORS

ACEMOGLU ET AL. (2015)

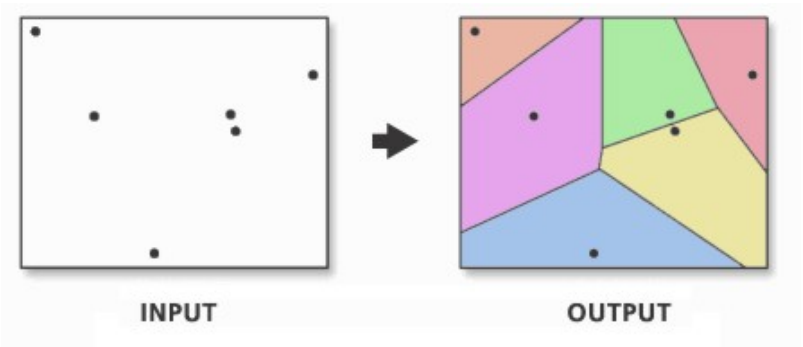
State capacity in one municipality

⇒ State capacity & prosperity in neighboring municipalities

- Polygon Neighbors: identify neighboring municipalities among 1017 in total in Colombia

CREATE THIESSEN POLYGONS

Divide surface by nearest point (aka Voronoi partition)



Alesina et al. (2016) use it as a robustness check to Murdock's ethnic homeland boundaries

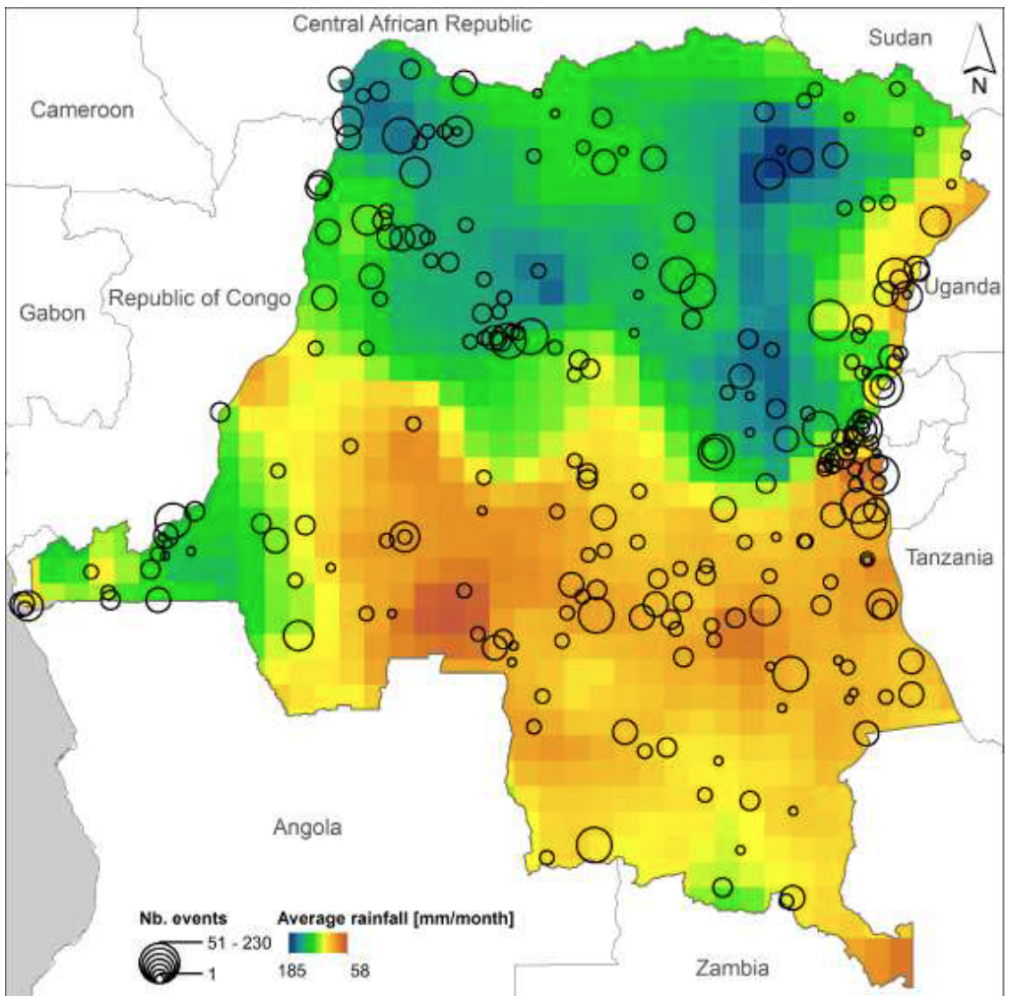
Alsan (2015) (cf. Lec 2) uses it as an alternative to the 20-miles radius of Ethnographic Atlas society location

KONIG ET AL. (2015)

Estimate the externality of military efforts on allied partners' effort in DR Congo civil wars

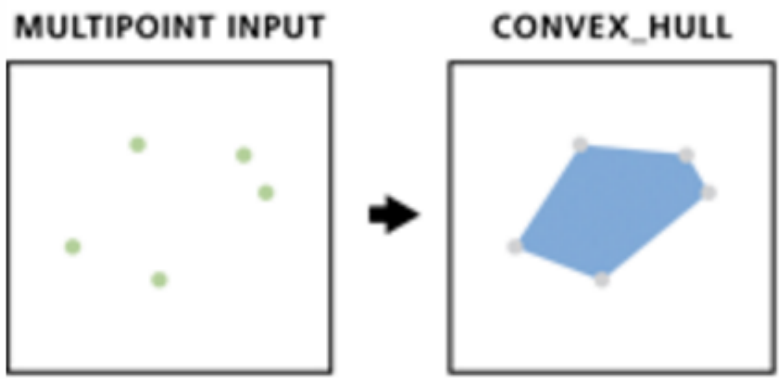
Use rainfall shock as instruments

But how do we measure the location of each military actor?



KONIG ET AL. (2015) (CONT.)

Minimum Bounding Geometry tool with Convex Hull option



Directional Distribution (Standard Deviation Ellipse) tool