The background features a detailed 3D rendering of a cityscape with buildings, roads, and infrastructure. A large globe is positioned on the right side, showing latitude and longitude lines. A red line graph with diamond markers is overlaid on the globe's surface, representing spatial data. The overall theme is spatial analysis and data visualization.

Spatial Statistics

Reference:

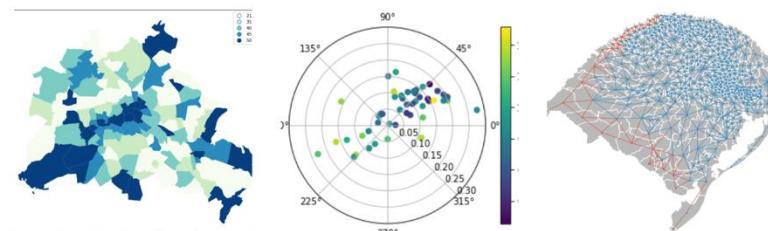
https://geographicdata.science/book/notebooks/04_spatial_weights.html#contiguity-weights

Capturing neighborhoods

- Spatial statistics between geographic tables
 - Often requires expensive computation, e.g. what's nearby?
 - Brute force = Pairwise distance calculation
 - Spatial weight matrices capture “topology” so that computation is cheaper

PySAL: Python Spatial Analysis Library

PySAL is an open source cross-platform library for geospatial data science with an emphasis on geospatial vector data written in Python.



[Housing Prices Berlin](#)

[Rose diagram \(directional LISAs\)](#)

[Visualizing Non Planar Neighbours](#)

PySAL supports the development of high level applications for spatial analysis, such as

- detection of spatial clusters, hot-spots, and outliers
- construction of graphs from spatial data
- spatial regression and statistical modeling on geographically embedded networks
- spatial econometrics
- exploratory spatio-temporal data analysis



PySAL

PySAL components

- **explore** - modules to conduct exploratory analysis of spatial and spatio-temporal data, including statistical testing on points, networks, and polygonal lattices. Also includes methods for spatial inequality, distributional dynamics, and segregation.
- **viz** - visualize patterns in spatial data to detect clusters, outliers, and hot-spots.
- **model** - model spatial relationships in data with a variety of linear, generalized-linear, generalized-additive, and nonlinear models.
- **lib** - solve a wide variety of computational geometry problems:
 - graph construction from polygonal lattices, lines, and points.
 - construction and interactive editing of spatial weights matrices & graphs
 - computation of alpha shapes, spatial indices, and spatial-topological relationships
 - reading and writing of sparse graph data, as well as pure python readers of spatial vector data.

Tobler's First Law of Geography:

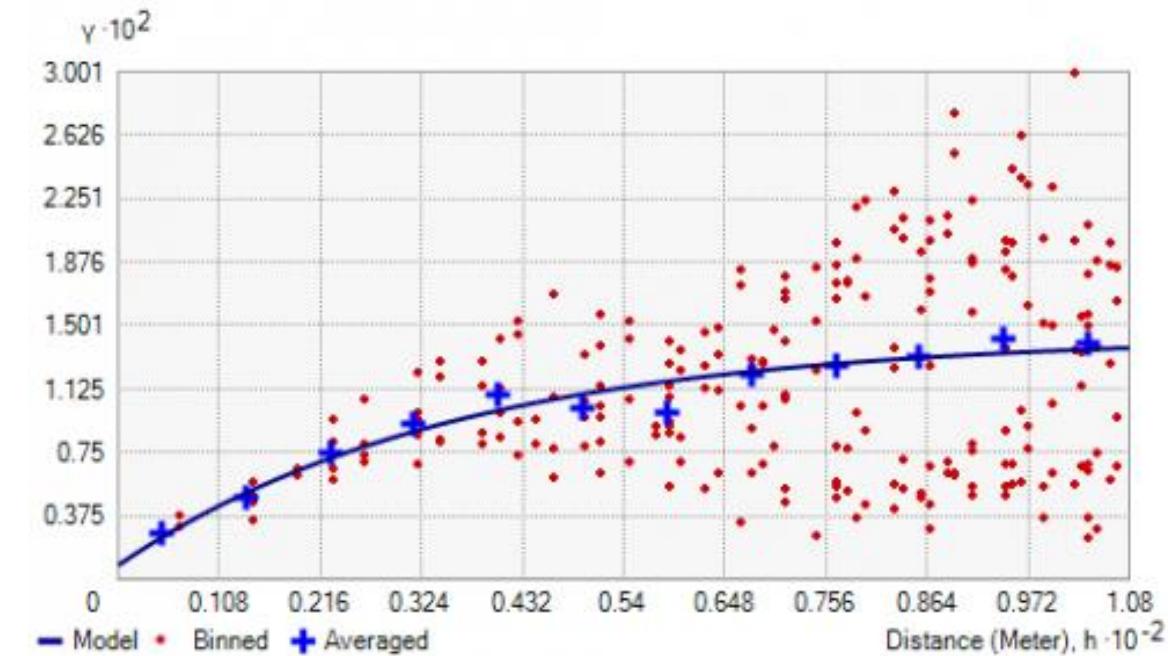
"Everything is related to everything else, but near things are more related than distant things."

Tobler, W. (1970) "A computer movie simulating urban growth in the Detroit region". Economic Geography, 46(2): 234-240.

Examples



Economic activity for shops nearby



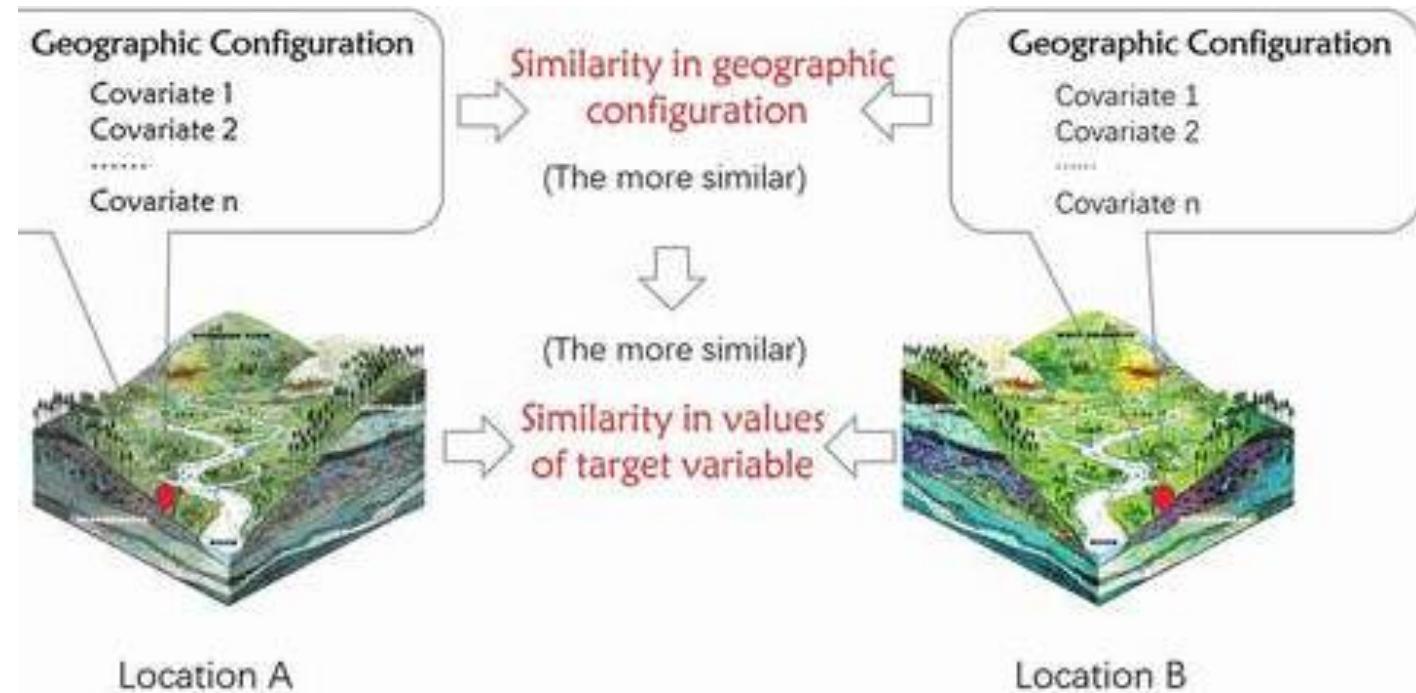
Terrain variability from a central point

The second law of geography

the phenomenon external to a geographic area of interest affects what goes on inside.



Third Law of Geography



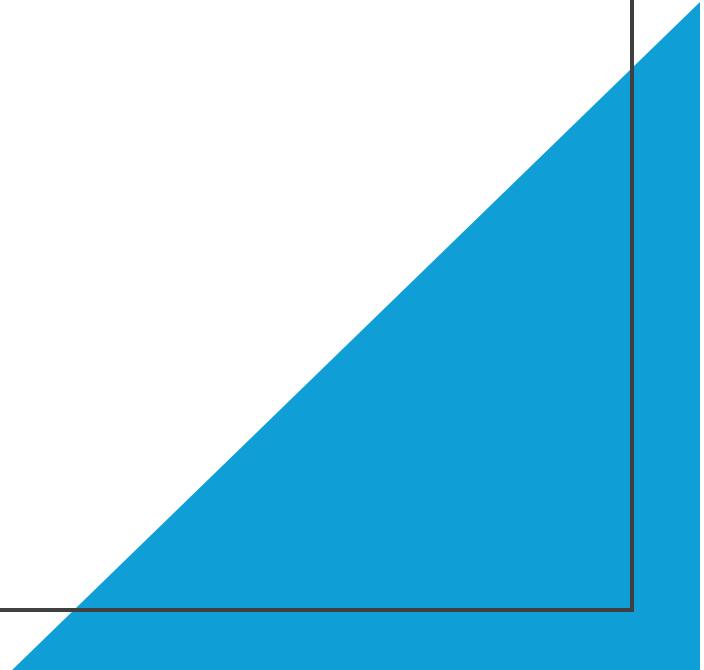
How to weigh your neighbors?

Continuity/Adjacency
relations

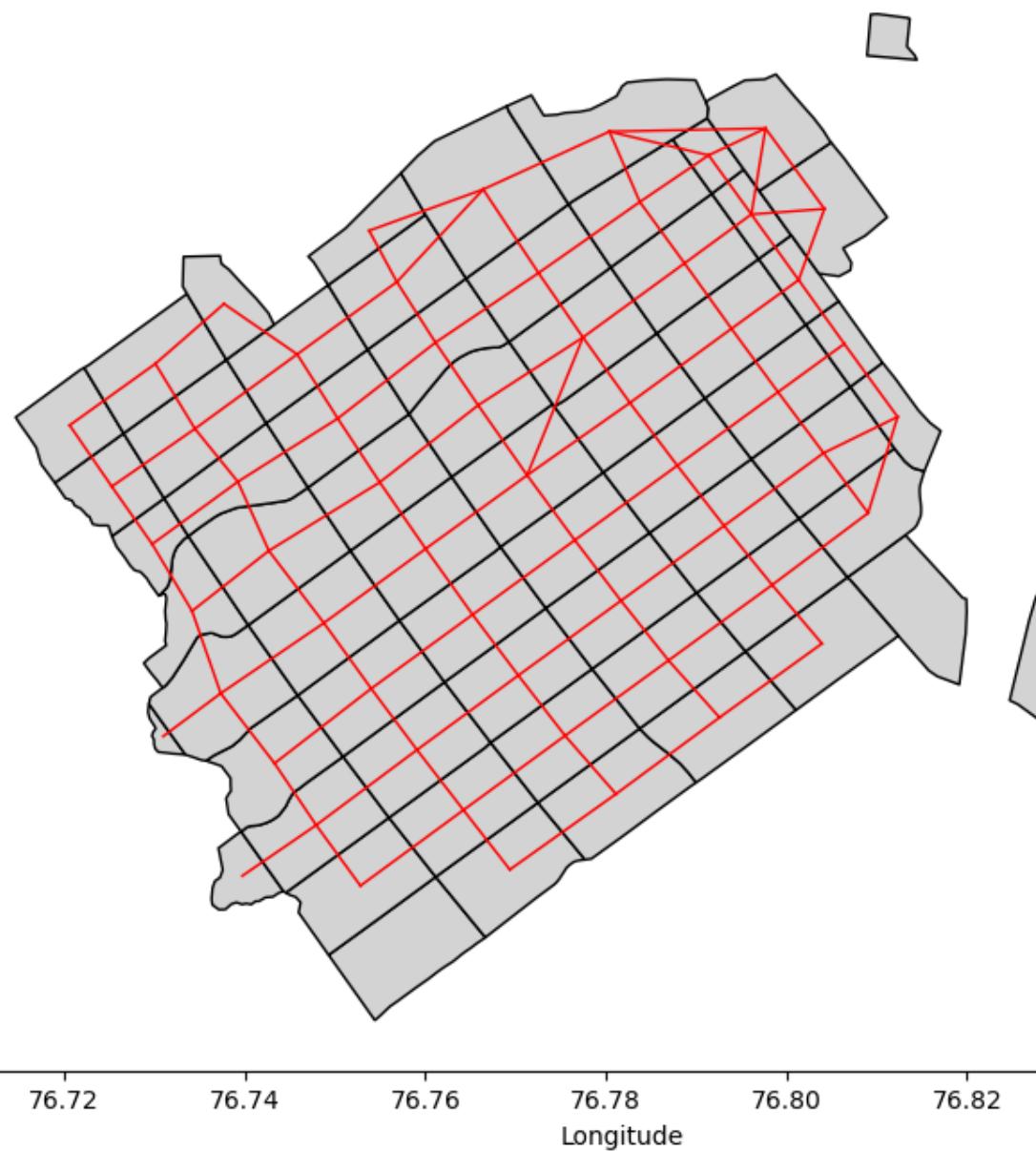
Distance based
relations

Hybrid weights

Continuity weights



Rook Contiguity Network for Chandigarh Sectors

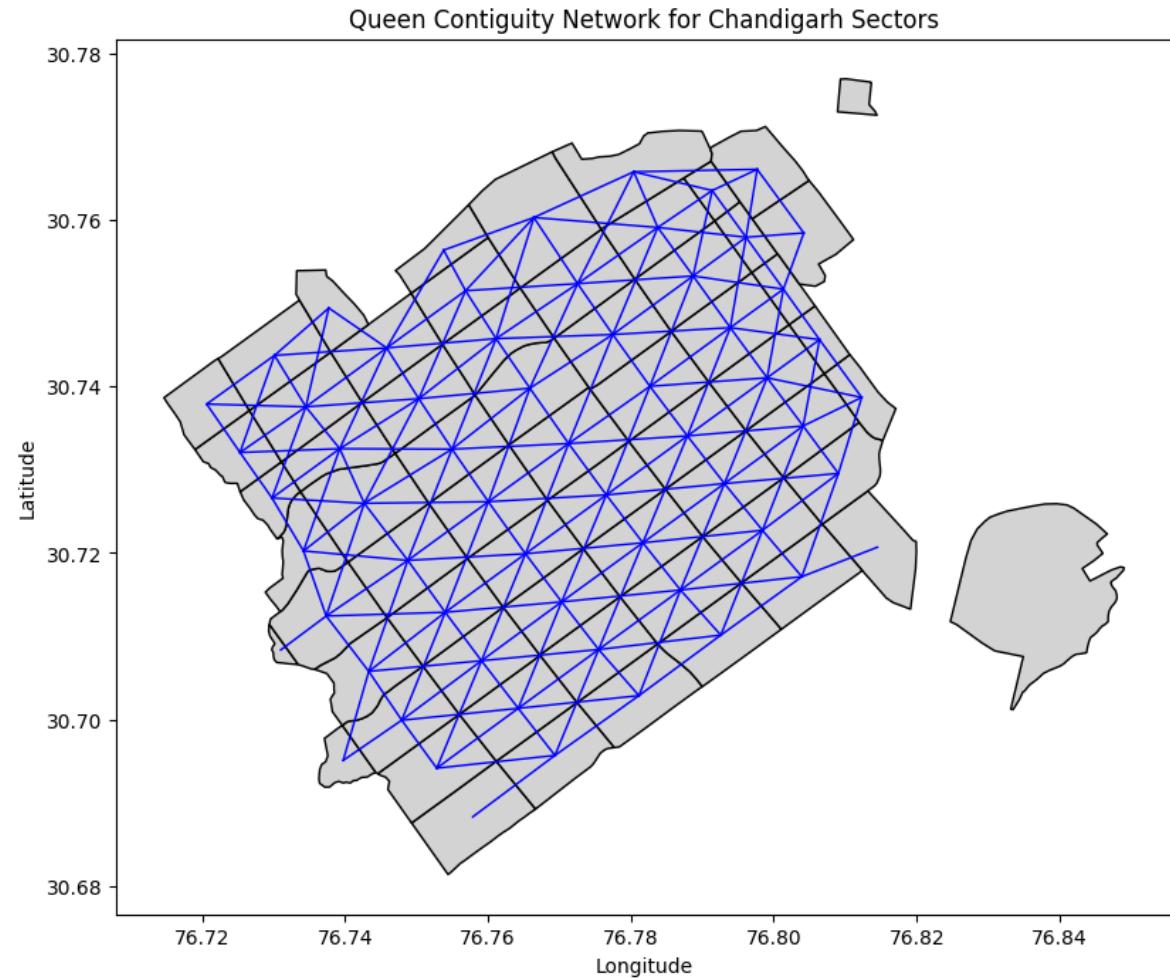


Rook Neighbors

- 4 islands
- Connections amongst sectors that share an edge
- See code example on notebook.
- Note, `weights.full()` gives matrix representation
- Useful for modeling more rigid phenomenon, e.g.,
 - Roads, zoning, water flows, etc.

Queen Contiguity

- Neighbors share an edge or a vertex
 - Diagonal movement is allowed
- Allows for a more flexible definition of neighborhood
 - Useful for modeling more ‘free’ phenomenon, e.g., trade, ecology, etc.



A note on islands

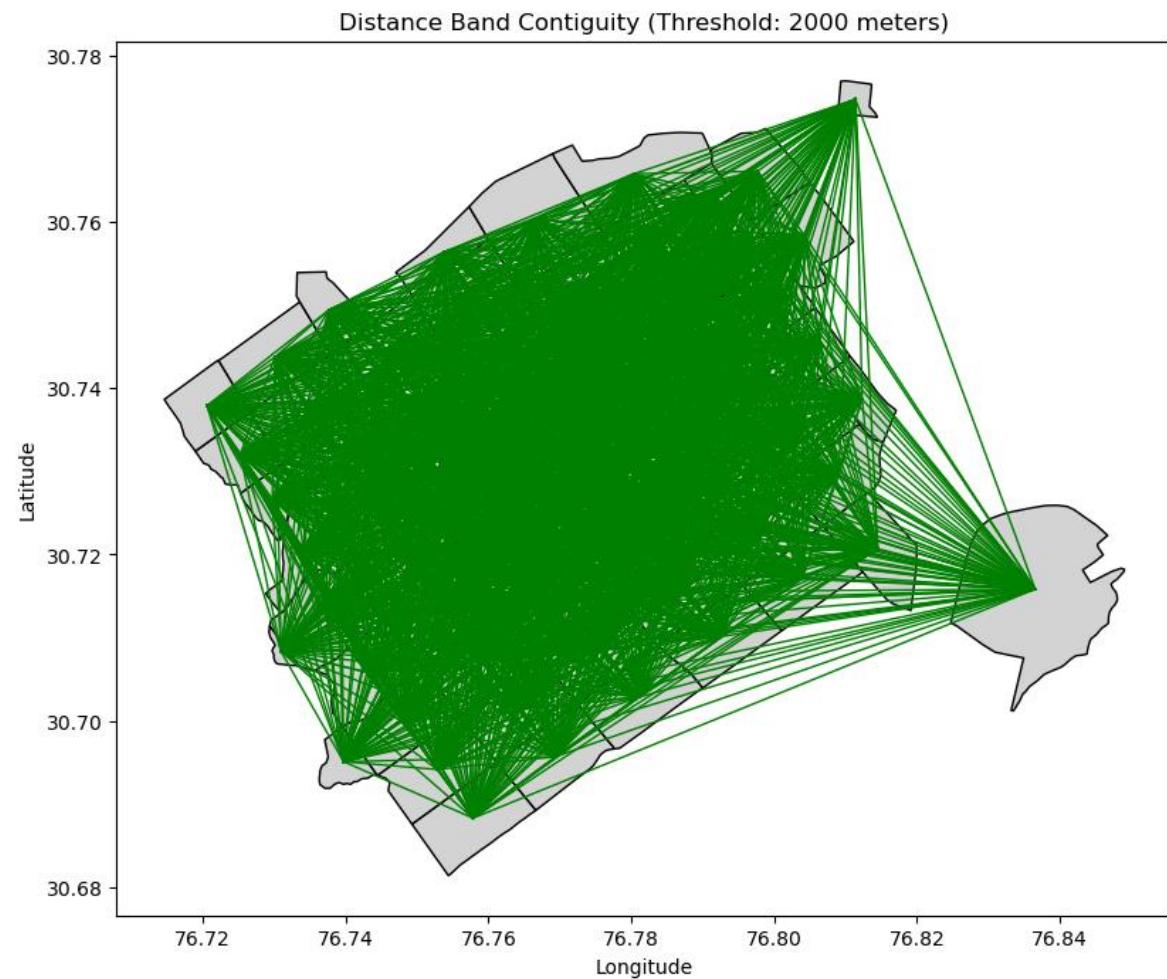
- Islands can create a problem in spatial statistics algorithms. It is thus preferred to add a single neighbor at least for further analysis.
 - Can be added by using sets (covered later) or
 - Can be added through manual editing of contiguity dictionary

Distance based relations



Define neighbors through a distance threshold

- Take all neighbors within a threshold distance
 - For example, blinkit deliveries
- Or just a few nearest neighbors,
 - E.g. kNN neighbors
 - Uses inter-centroid distances
- Shall we just use kernels?
 - Shape, e.g., gaussian, triangular,
 - Bandwidth, e.g., 500m. After this the weights are decayed.



Different kernel functions

$$z_{i,j} = d_{i,j}/h_i$$

triangular

$$K(z) = (1 - |z|) \text{ if } |z| \leq 1$$

uniform

$$K(z) = 1/2 \text{ if } |z| \leq 1$$

quadratic

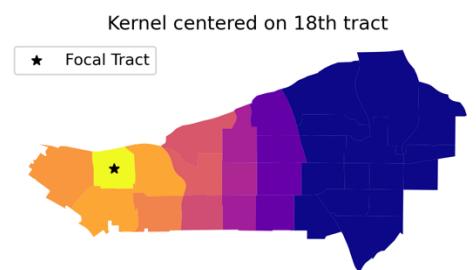
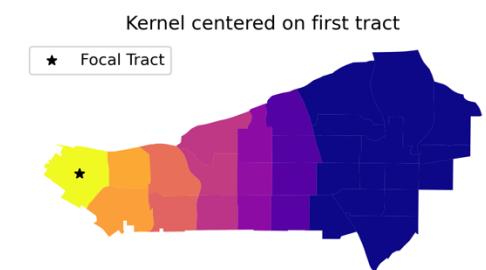
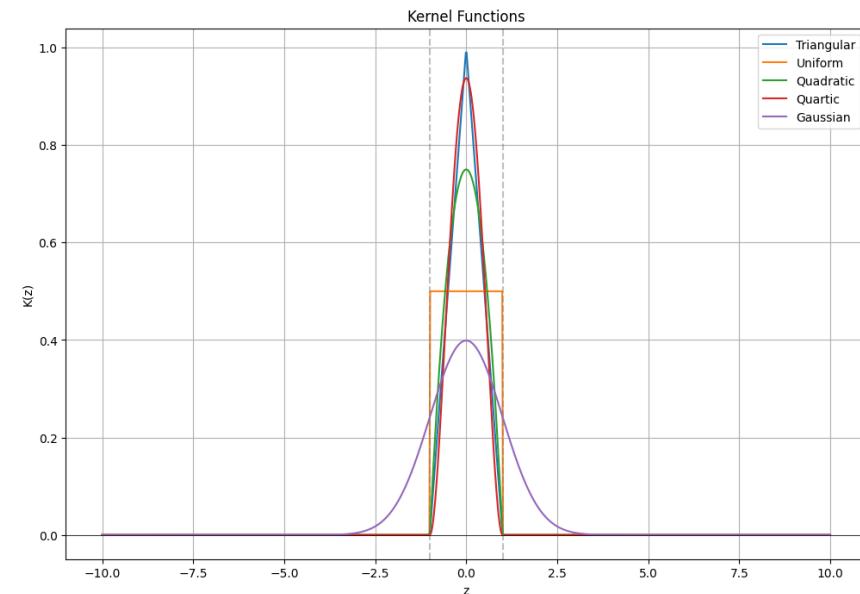
$$K(z) = (3/4)(1 - z^2) \text{ if } |z| \leq 1$$

quartic

$$K(z) = (15/16)(1 - z^2)^2 \text{ if } |z| \leq 1$$

gaussian

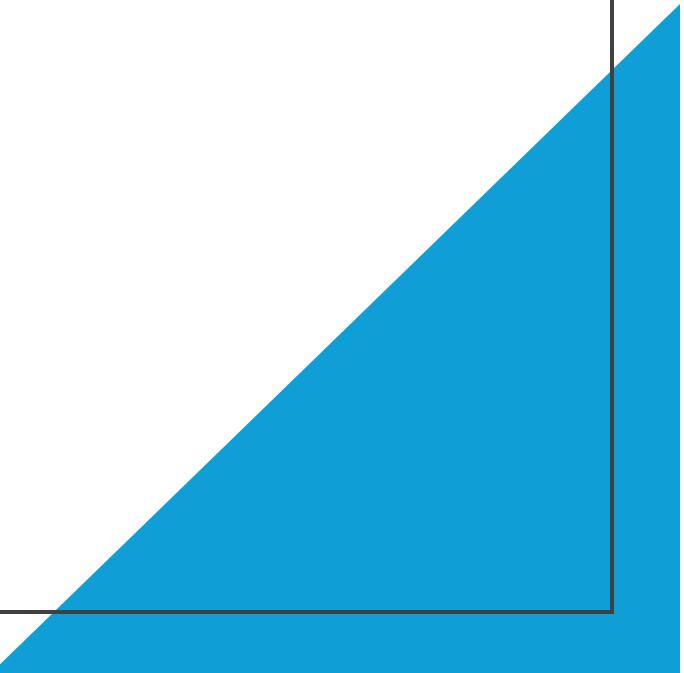
$$K(z) = (2\pi)^{-1/2} \exp(-z^2/2)$$



Summary

Spatial Weights Type	Best When...	Pros	Cons
Rook Contiguity	Influence spreads via shared borders (e.g., political units)	Simple, interpretable	Ignores diagonal connections
Queen Contiguity	Influence spreads via borders & corners	More inclusive than Rook	Still ignores distance
Distance Band	You have a fixed range of influence (e.g., air pollution, traffic impact)	Ensures all close units are neighbors	Sharp cutoff—no gradual influence
Inverse Distance	Influence decays with distance (e.g., economic interactions)	Accounts for gradual influence loss	Needs careful choice of power
K-Nearest Neighbors (KNN)	You need a fixed number of neighbors per unit	Ensures every unit has neighbors	No distance-based weighting
Kernel Weights	Influence spreads smoothly over distance	Most flexible, smooth decay	Harder to interpret

Hybrid



Hybrid weights

- Decaying distance within a threshold
 - `w_bdb = weights.distance.DistanceBand.from_dataframe(gdf, 1.5, binary=True)`
 - DistanceBand uses inverse distance relationship

Incorporating earth's curvature

```
# ignore curvature of the earth  
knn4_bad = weights.distance.KNN.from_shapefile(  
    "../data/texas/texas.shp", k=4  
)
```



```
radius = geometry.sphere.RADIUS_EARTH_MILES  
radius
```

```
knn4 = weights.distance.KNN.from_shapefile(  
    "../data/texas/texas.shp", k=4, radius=radius  
)
```

Block Weights



Block weights

- Use a list as proxy for near-ness, e.g., all sectors with a waste treatment plant

Combining block and distance?

- Define neighbors as *all schools in a ward within a distance of 2 km*
 - Use **Sets**
 - Sets allow you to do set operations - union, intersection, etc. of neighbors from different weight matrices

```
w_fixed_sets = weights.set_operations.w_union(w_rook, wk1)
```

In Research

Jia, Yuhao, Zile Wu, Shengao Yi, and Yifei Sun. "GeoTransformer: Enhancing Urban Forecasting with Geospatial Attention Mechanisms." *arXiv preprint arXiv:2408.08852* (2024).

- <https://arxiv.org/abs/2408.08852>

Zhu, Di, Yu Liu, Xin Yao, and Manfred M. Fischer. "Spatial regression graph convolutional neural networks: A deep learning paradigm for spatial multivariate distributions." *GeoInformatica* 26, no. 4 (2022): 645-676.

<https://link.springer.com/article/10.1007/s10707-021-00454-x>

Liu, Pengyuan, and Filip Biljecki. "A review of spatially-explicit GeoAI applications in Urban Geography." *International Journal of Applied Earth Observation and Geoinformation* 112 (2022): 102936.

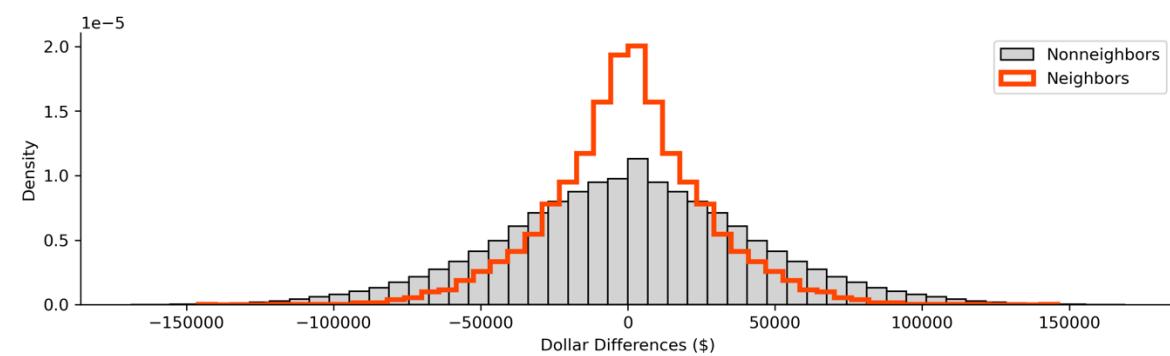
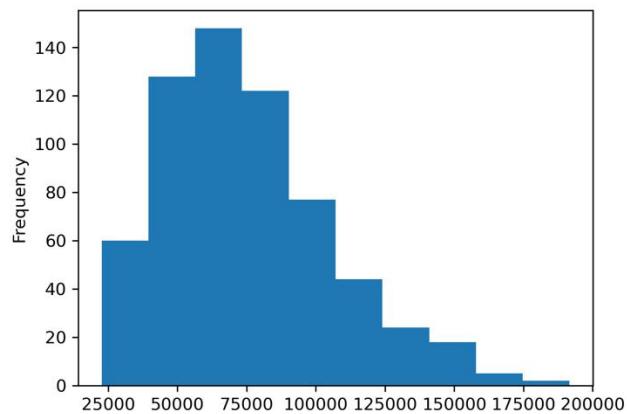
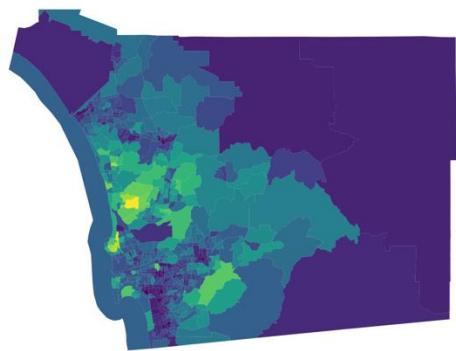
<https://www.sciencedirect.com/science/article/pii/S1569843222001339>

Zhao, Tianhong, Xiucheng Liang, Wei Tu, Zhengdong Huang, and Filip Biljecki. "Sensing urban soundscapes from street view imagery." *Computers, Environment and Urban Systems* 99 (2023): 101915.

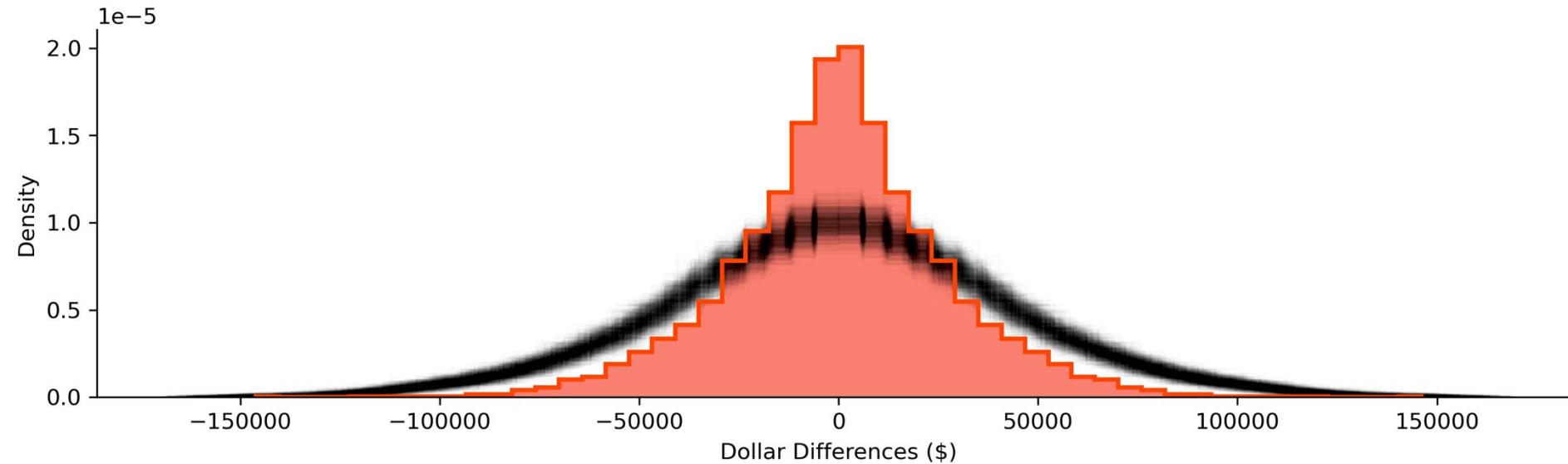
What do we do with neighbors?

Example use cases

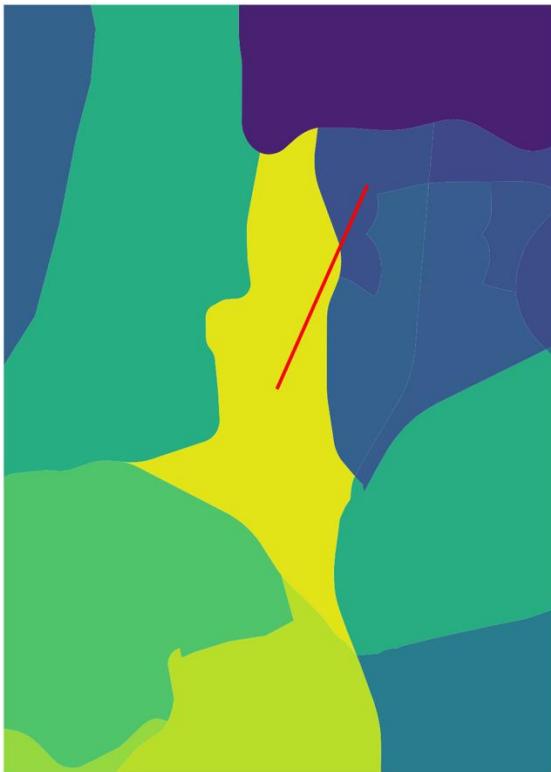
Data variation in neighbors



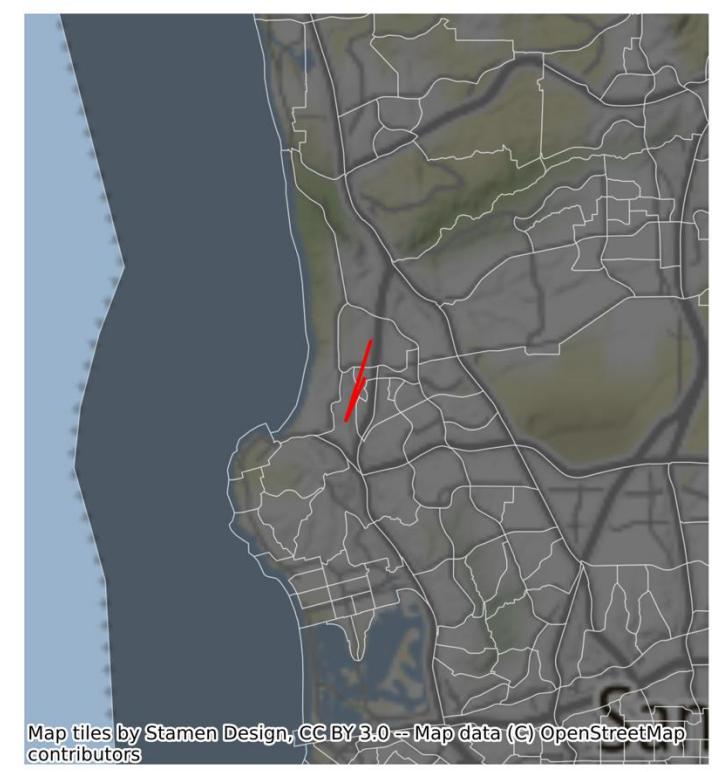
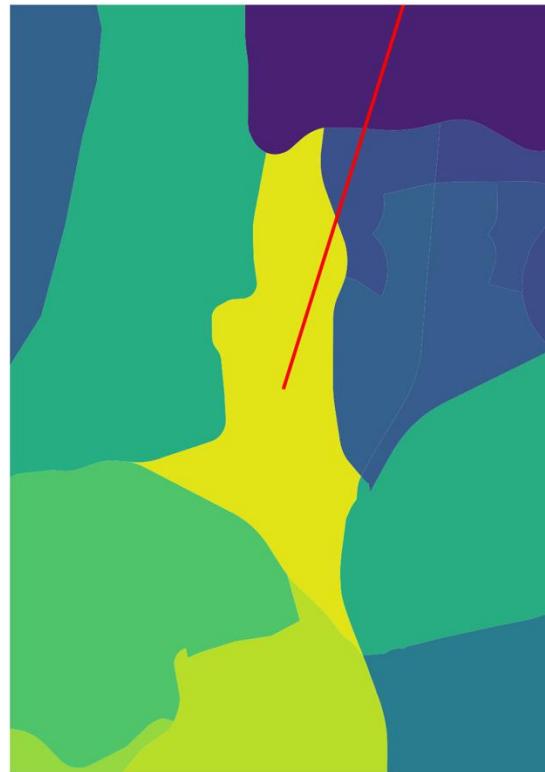
Comparison against random distribution of incomes



Finding the outliers: Extreme differences



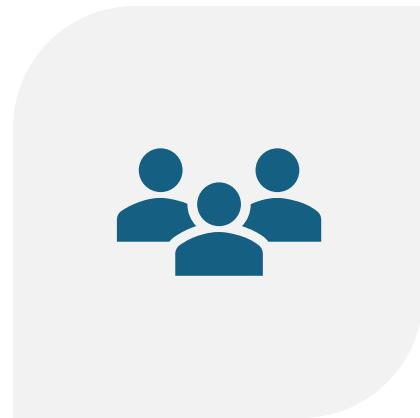
- Potentially useful for flagging anomalies, inequities, artificially inflated prices or agents of artificial change (e.g., increase pollution levels, etc.)



Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

Data exploration through Choropleths

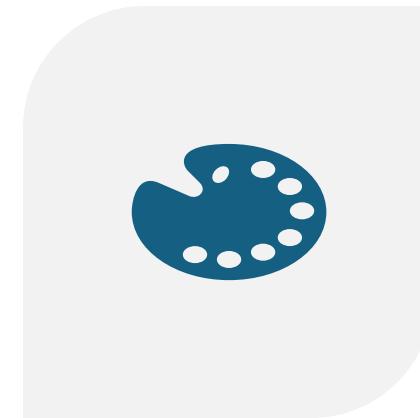
The steps



IDENTIFY NUMBER OF
GROUPS



FIND A CLASSIFICATION
METHOD



SELECT A COLOR
SCHEME

Classifying data

Equal interval classification

```
ei5 = mapclassify.EqualInterval(mx["PCGDP1940"], k=5)
```

Quantiles

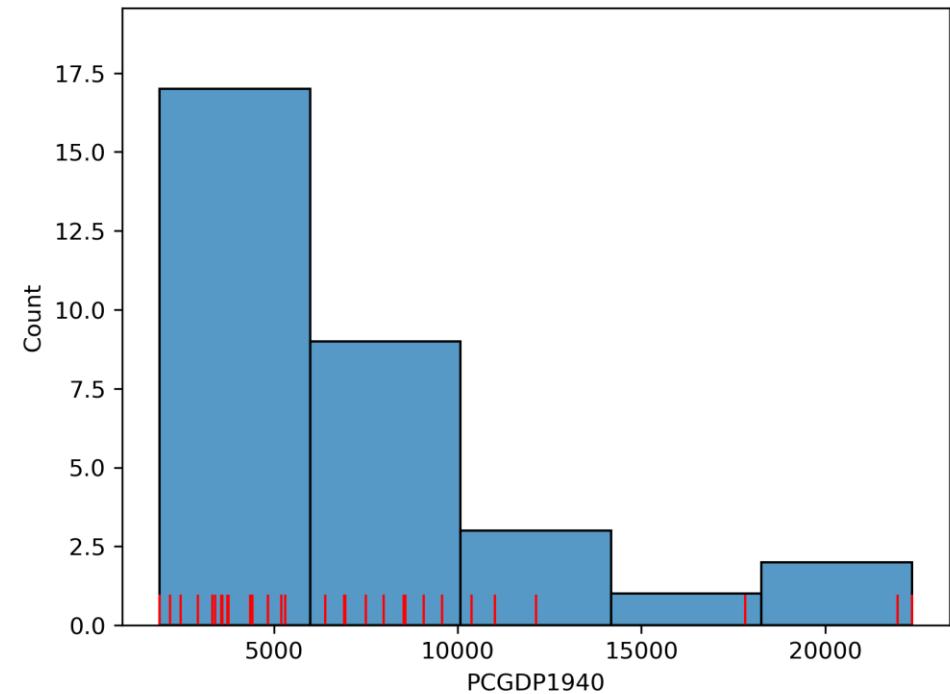
```
q5 = mapclassify.Quantiles(mx.PCGDP1940, k=5)
```

Using mean and std deviation, K sigma boundaries

```
msd = mapclassify.StdMean(mx["PCGDP1940"])
```

Ensure class separation

```
mb5 = mapclassify.MaximumBreaks(mx["PCGDP1940"], k=5)
```



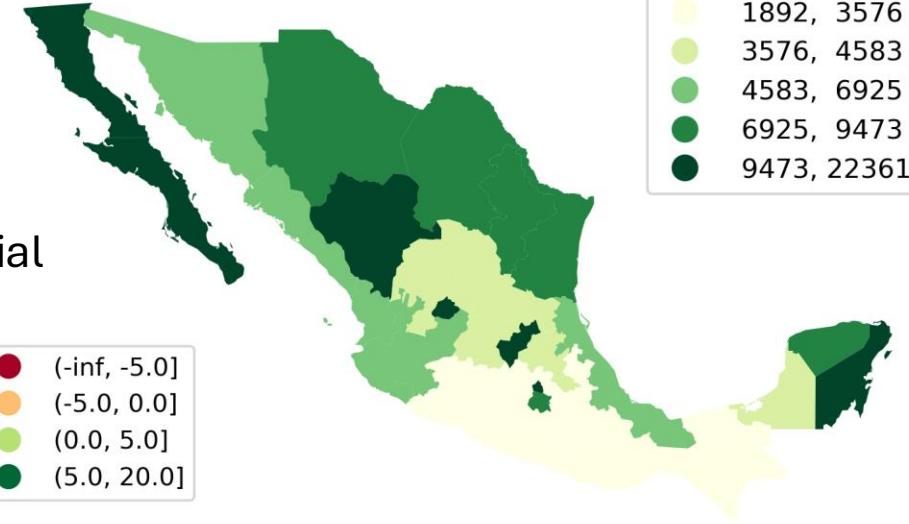
Observe left skew in data

Choosing a color

- Sequential
- Diverging
- Qualitative



Sequential



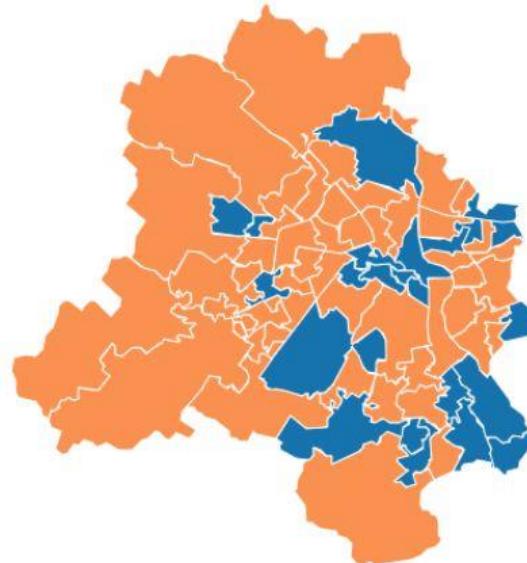
Diverging



Qualitative

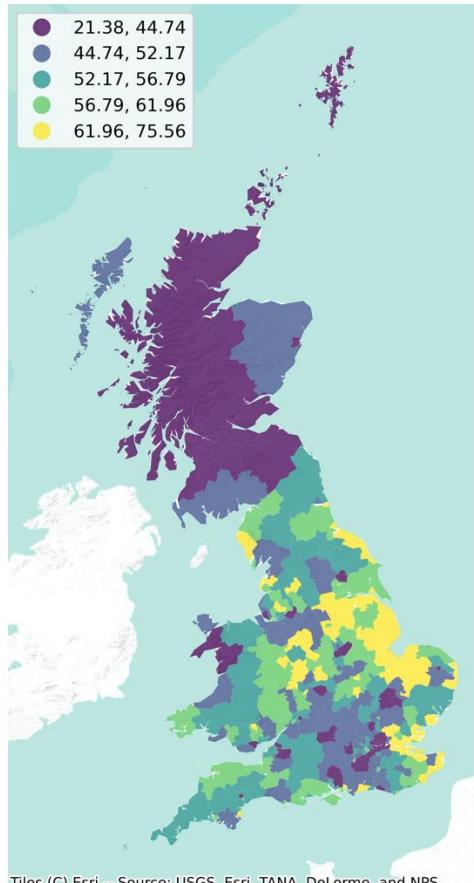
A recent choropleth

Party	Won	Leading	Total
Bharatiya Janata Party - BJP	14	33	47
Aam Aadmi Party - AAP	11	12	23
Total	25	45	70



Global spatial autocorrelation

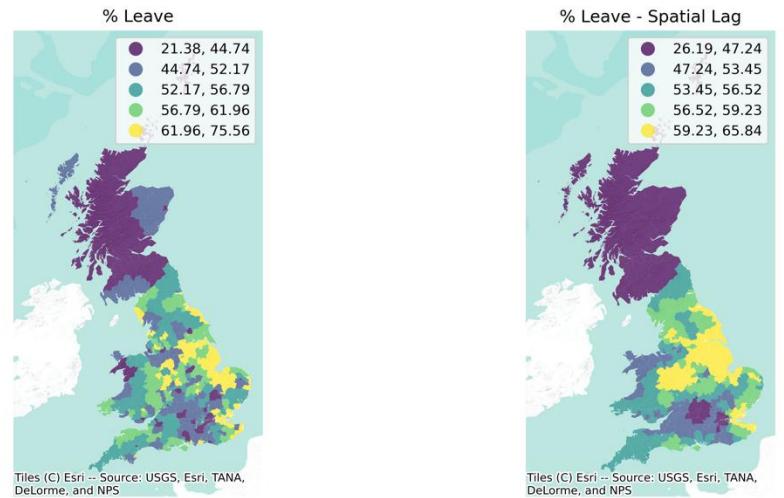
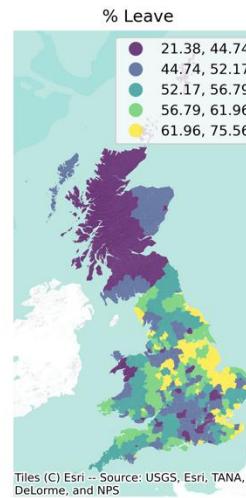
Spatial lag



Quantile plot of the Brexit vote

$$Y_{sl} = \mathbf{W}Y$$

Output at every shape
(weighed sum of
neighbors)



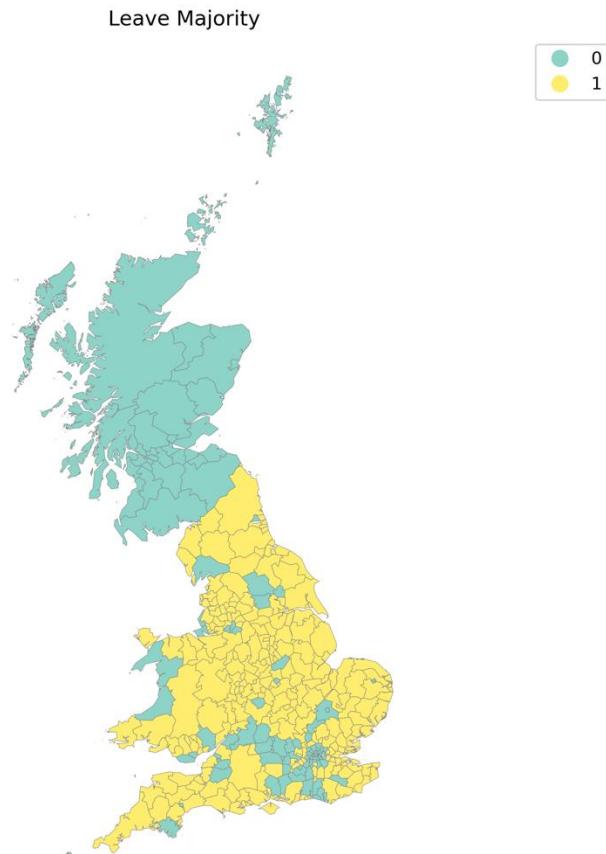
The blurring effect
of running an
averaging kernel

	Pct_Leave	Pct_Leave_lag
lad16cd		
E08000012	41.81	54.61375
S12000019	37.94	38.01875

Wants to stay but
neighbors want to leave

Uniform to all its
neighbors

Join counts



Binarized decisions

Count number of GY, GG, YY joins

Find statistical significance through generated random sequences

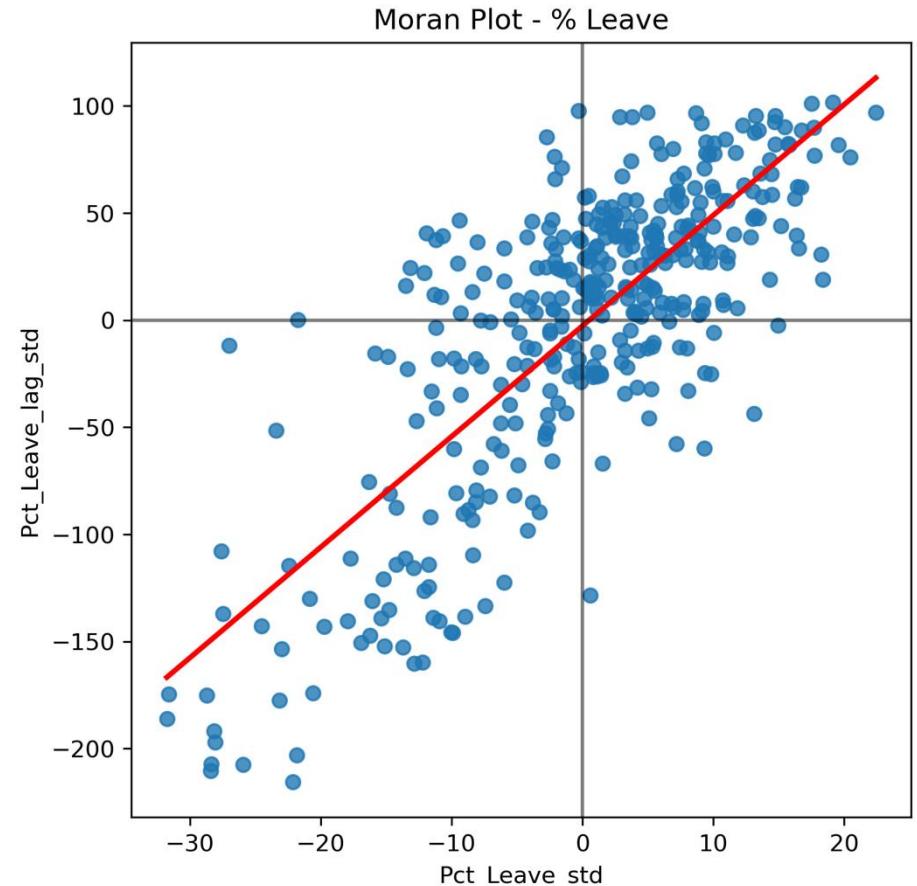
Moran's Plot

Analyzing how related is a variable to its spatial lag through a scatter plot

$$I = \frac{n}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij} z_i z_j}{\sum_i z_i^2}$$

Moran's I. z is the standardized variable and w is the weights

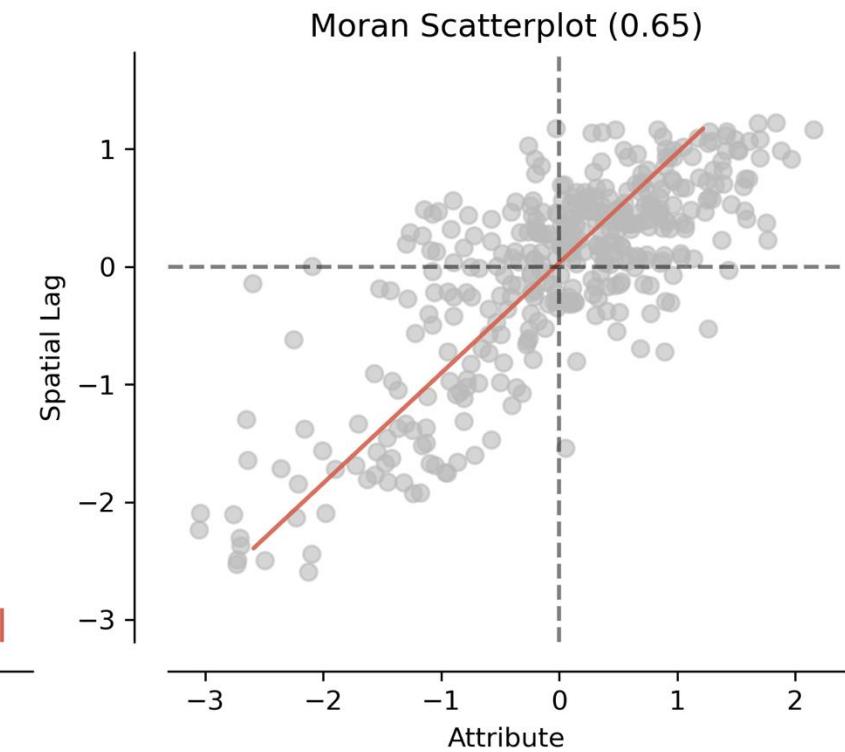
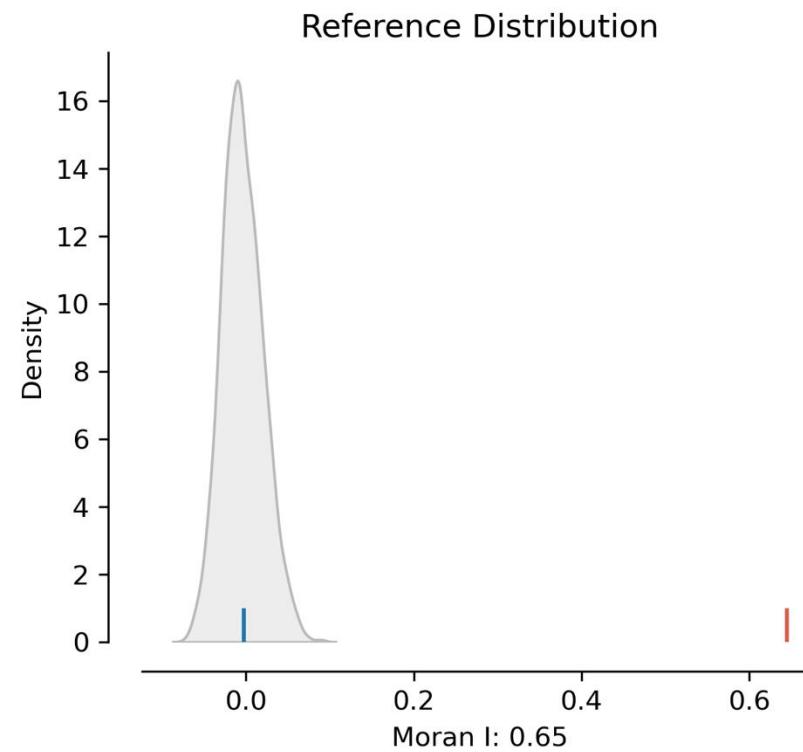
Indicates that spatial correlation is high.
Low values are around other low values and vice versa. Moran's I is the slope of this line.



Check for statistical significance

Results fromesda simulation

A legend box containing two items:
- A green line segment followed by the text "moran.p_sim"
- A grey line segment followed by the text "0.001"





Example 1: Disease Outbreaks

- **Scenario:** A public health researcher is studying the spread of dengue fever in a city.
- **Application:** The researcher calculates Moran's I to see if dengue cases are clustered, randomly distributed, or dispersed.
- **Interpretation:**
 - If Moran's I is significantly positive, dengue cases are spatially clustered, possibly due to localized mosquito breeding grounds.
 - If Moran's I is near zero, the distribution of cases resembles a random pattern.
 - If Moran's I is significantly negative, cases are dispersed, suggesting strong spatial inhibition (unlikely in this context).

Example 2: Crime patterns

- **Scenario:** A police department is analyzing burglary reports across different districts.
- **Application:** Moran's I helps determine if crime is clustered, random, or dispersed. Ripley's K can be used to analyze clustering at different spatial scales.
- **Interpretation:**
 - High Moran's I → Crimes are clustered, possibly due to socioeconomic or environmental factors.
 - Low Moran's I → Crimes are randomly distributed, suggesting no significant spatial pattern.



How does the Moran's I change here?

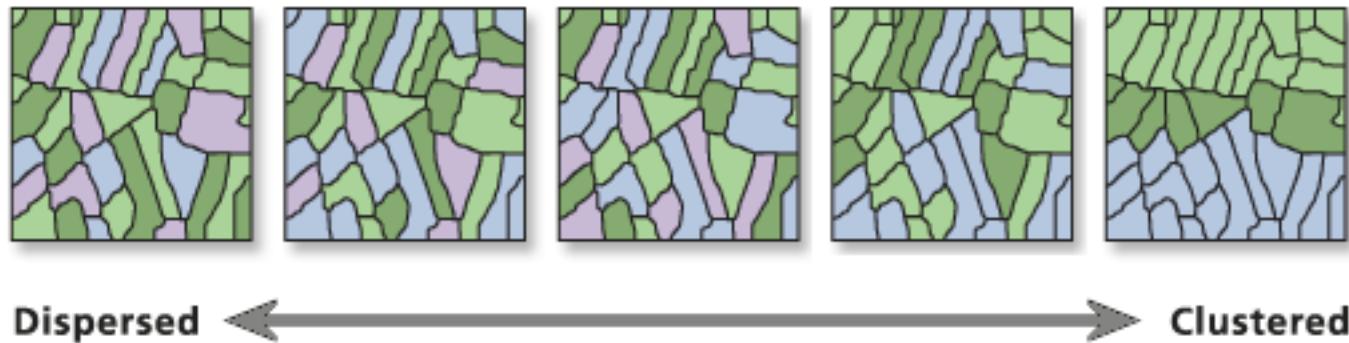


Image ref:

<https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/spatial-autocorrelation.htm>

Local Spatial Autocorrelation

Global vs Local

Global

- Determines whether the overall spatial distribution is compatible with geographic random process
- Could uncover processes that generate association between values
 - Spillovers
 - Contagion
- Could uncover data measurement errors
- They are primarily a single measure of the entire map
 - Hard to find outliers
 - Easier to establish global phenomenon

Local

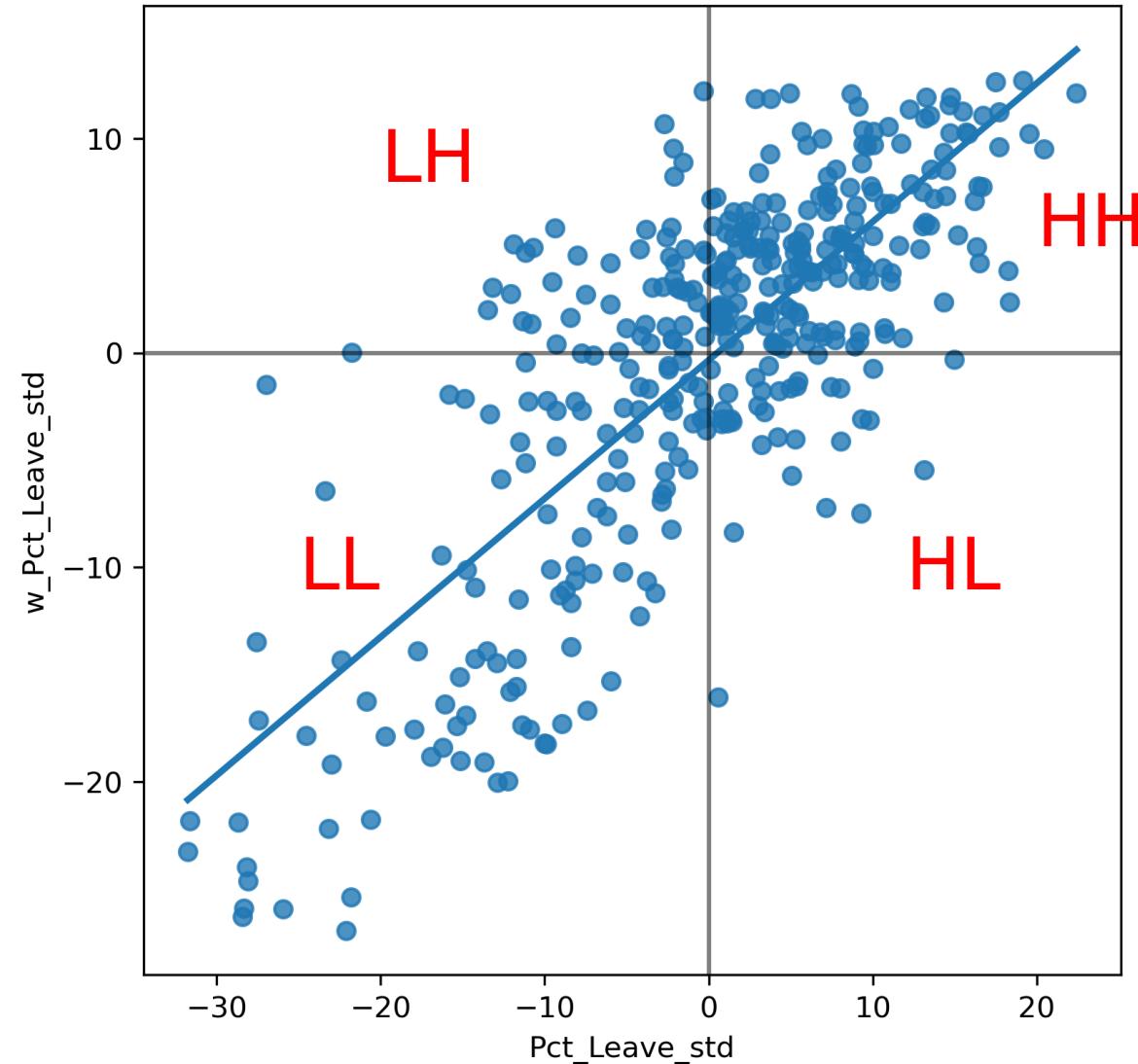
- Focus on each observation and its surroundings
- These are scores, not summaries
 - For example, Local Indicators of Spatial Association (LISAs)

Let's start dissecting the Morgan's I

- Above average leave voting
- Below average leave voting

and

- Above/Below average lag
with neighbors



Local Moran's I (A LISA Statistic)

- Where is the unusual concentration of values?
- Does this location belong to a statistically significant cluster?

$$I_i = \frac{z_i}{m_2} \sum_j w_{ij} z_j ; m_2 = \frac{\sum_i z_i^2}{n}$$

where m_2 is the second moment (variance) of the distribution of values in the data, $z_i = y_i - \bar{y}$, $w_{i,j}$ is the spatial weight for the pair of observations i and j , and n is the number of observations.

Properties of Local Moran's I



Negative and positive deviations **from the mean** are identically handled



Higher I amongst low deviations
is less significant than higher I
amongst high deviations

Conformance in extremes is more rare



An easy identifier for anomalies/clusters in the data

Examples of LISA usage



GEOGRAPHIC CLUSTERS OF
POVERTY

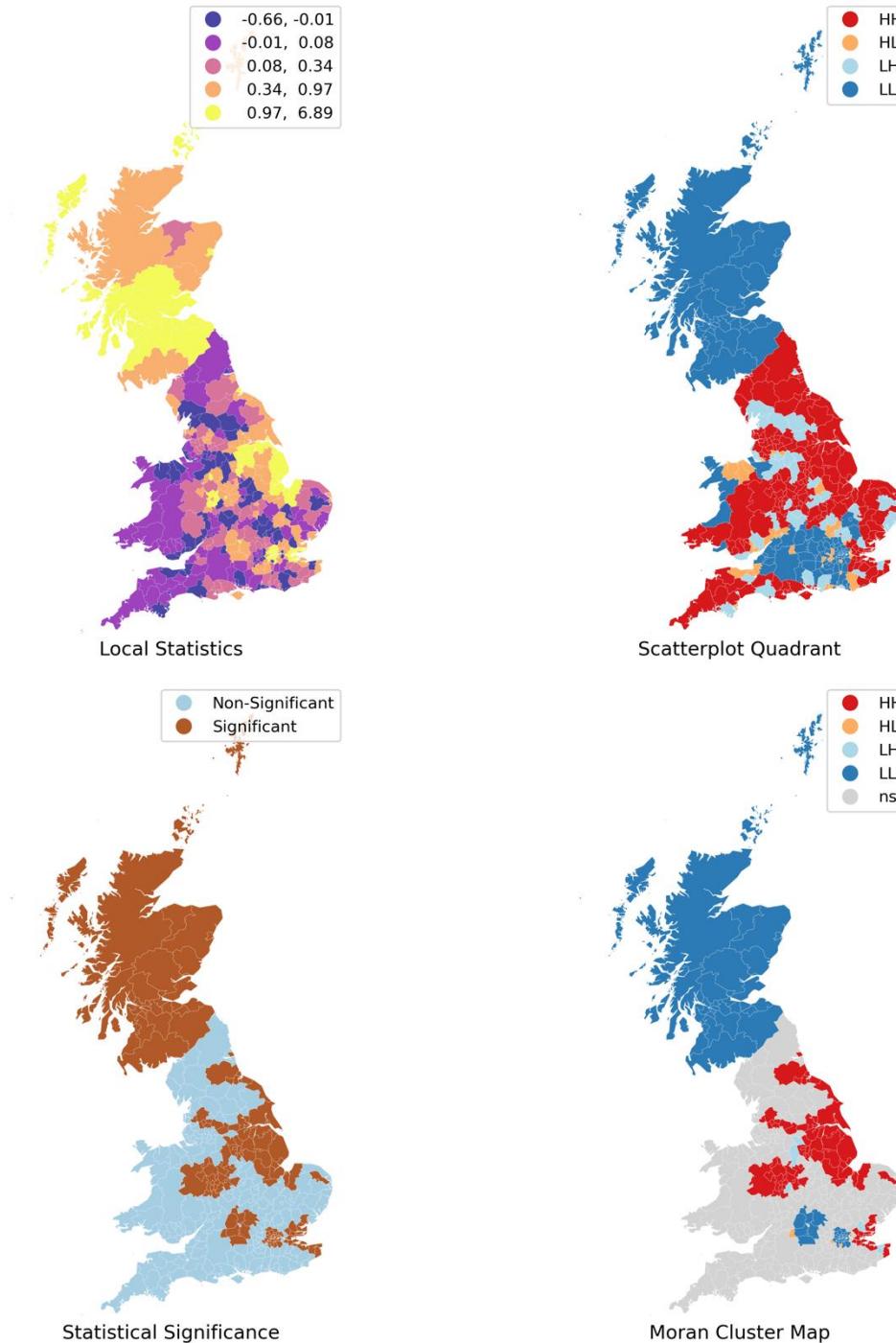


CLUSTERS OF CONTAGIOUS
DISEASES



AREAS OF PARTICULARLY
HIGH/LOW ECONOMIC ACTIVITY

Interpreting LISAs



HH

- Hot spots

LL

- Cold Spots

LH

- Doughnuts
- Low values surrounded by high values

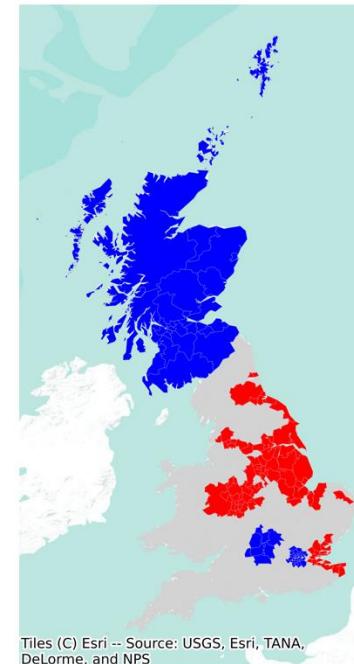
HL

- Diamonds in the rough

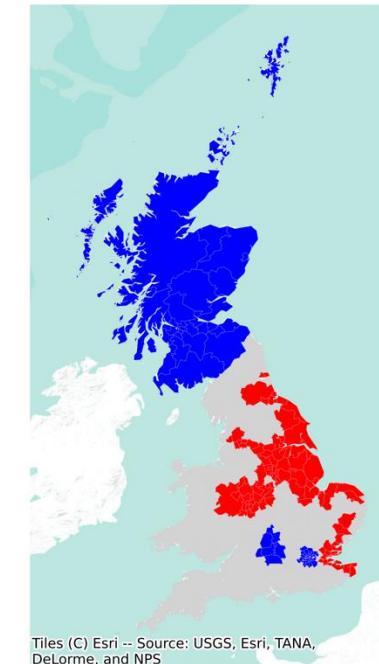
Getis' Local Statistic

- G_i statistic – excludes i
- G^*_i statistic – includes i
- G statistics only allow identification of positive correlation
 - Positive values – high values together
 - Negative values – low values together

G statistic for Pct of Leave votes



G^* statistic for Pct of Leave votes



Use G and I together

- Moran's cluster highlights all types of correlation
 - Highly high values surrounded by high values (HH) or low values surrounded by low values (LL)
 - Even flags outliers (LH, HL)
- G statistic informs of the hotspots and cool spots
 - Consistently high/low values



Moran Cluster Map

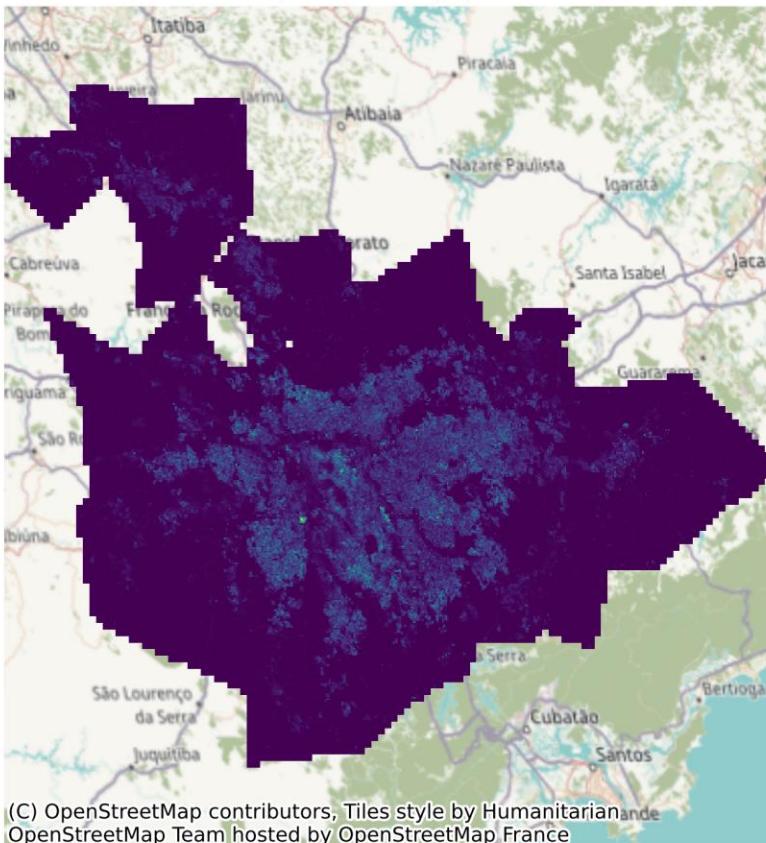
G statistic for Pct of Leave votes



Tiles (C) Esri -- Source: USGS, Esri, TANA, DeLorme, and NPS

Application to surfaces

Population by pixel



Population clusters

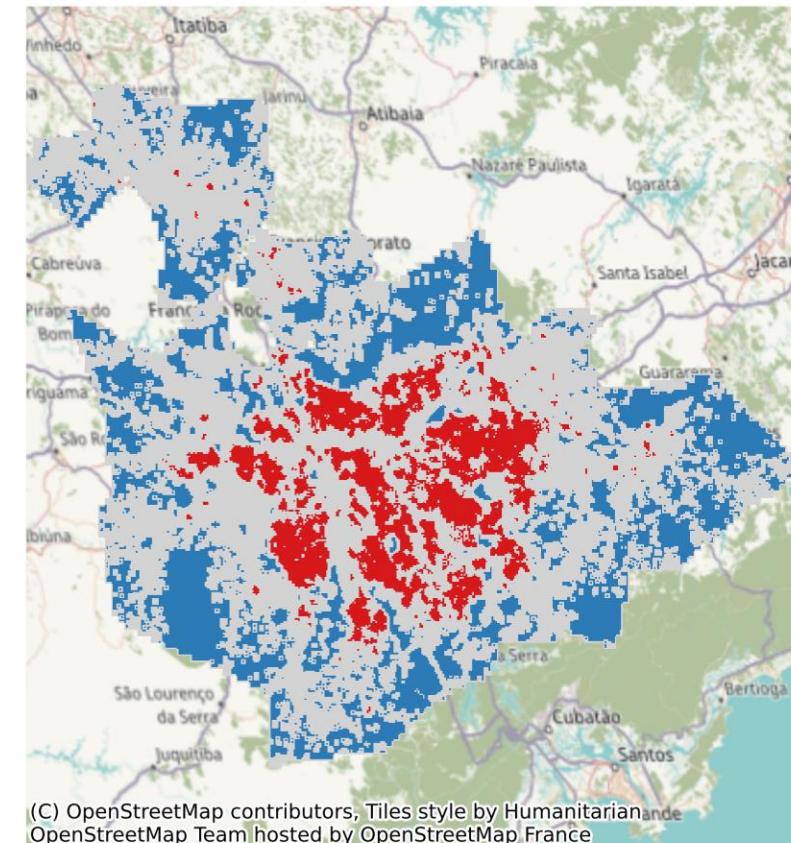
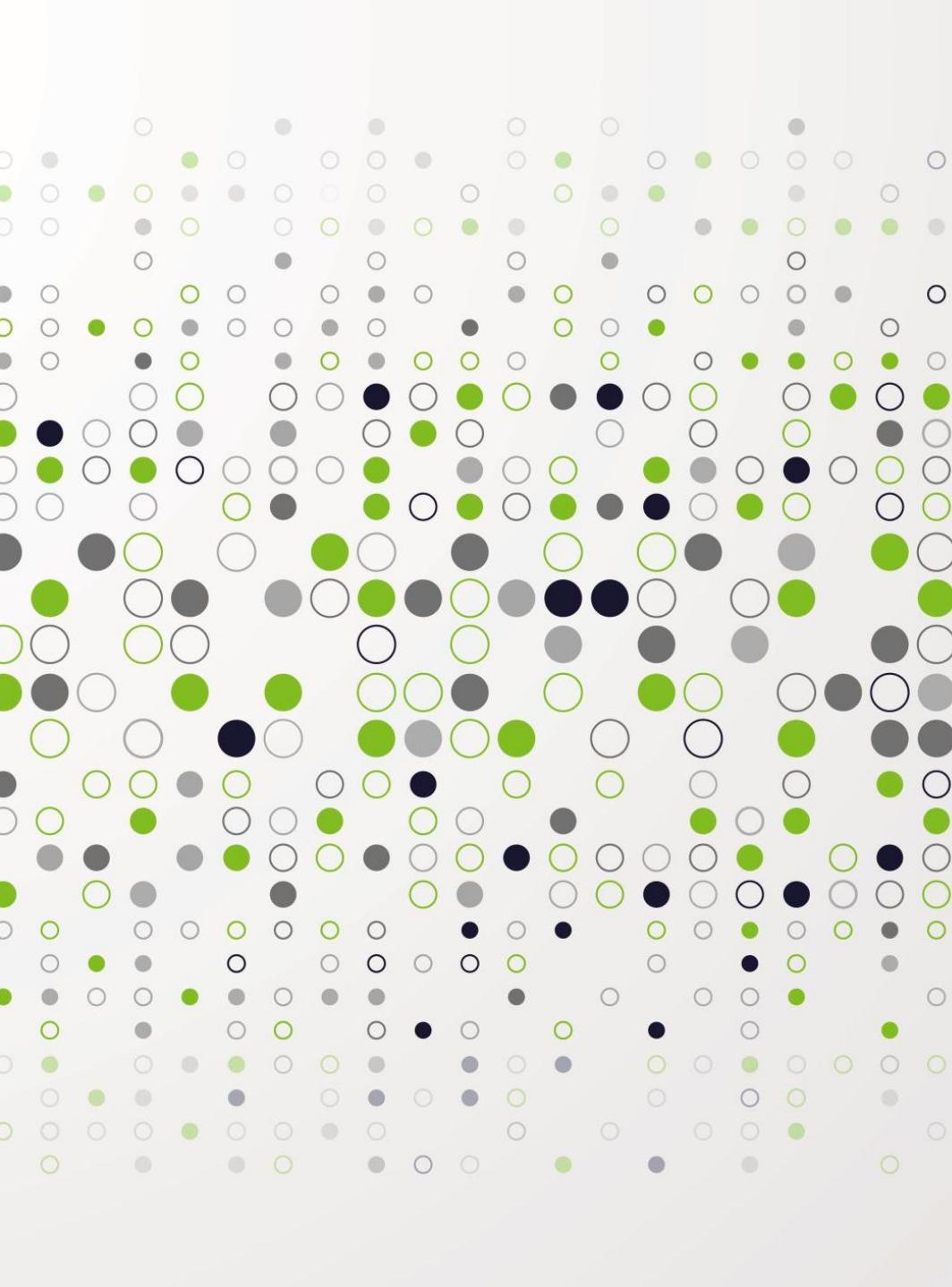




Image: <https://techspace.hashnode.dev/visualizing-geospatial-data-using-kepler-gl>



Point pattern?

- The pattern is more important than the point
 - Measuring number of cars
 - “When points are seen as events that could take place in several locations but only happen in a few of them, a collection of such events is called a *point pattern*.”
- Captures an underlying geographical process



Point pattern analysis

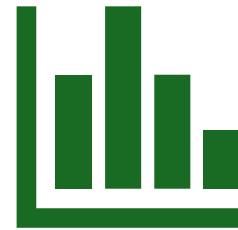
- *What does the pattern look like?*
- *What is the nature of the distribution of points?*
- *Is there any structure in the way locations are arranged over space? That is, are events clustered? or are they dispersed?*
- *Why do events occur in those places and not in others?*

Reminder: Probabilistic vs Statistical Thinking



Probabilistic

Frequentist
Bayesian



Statistical

Observe data to understand underlying
processes that generate the data

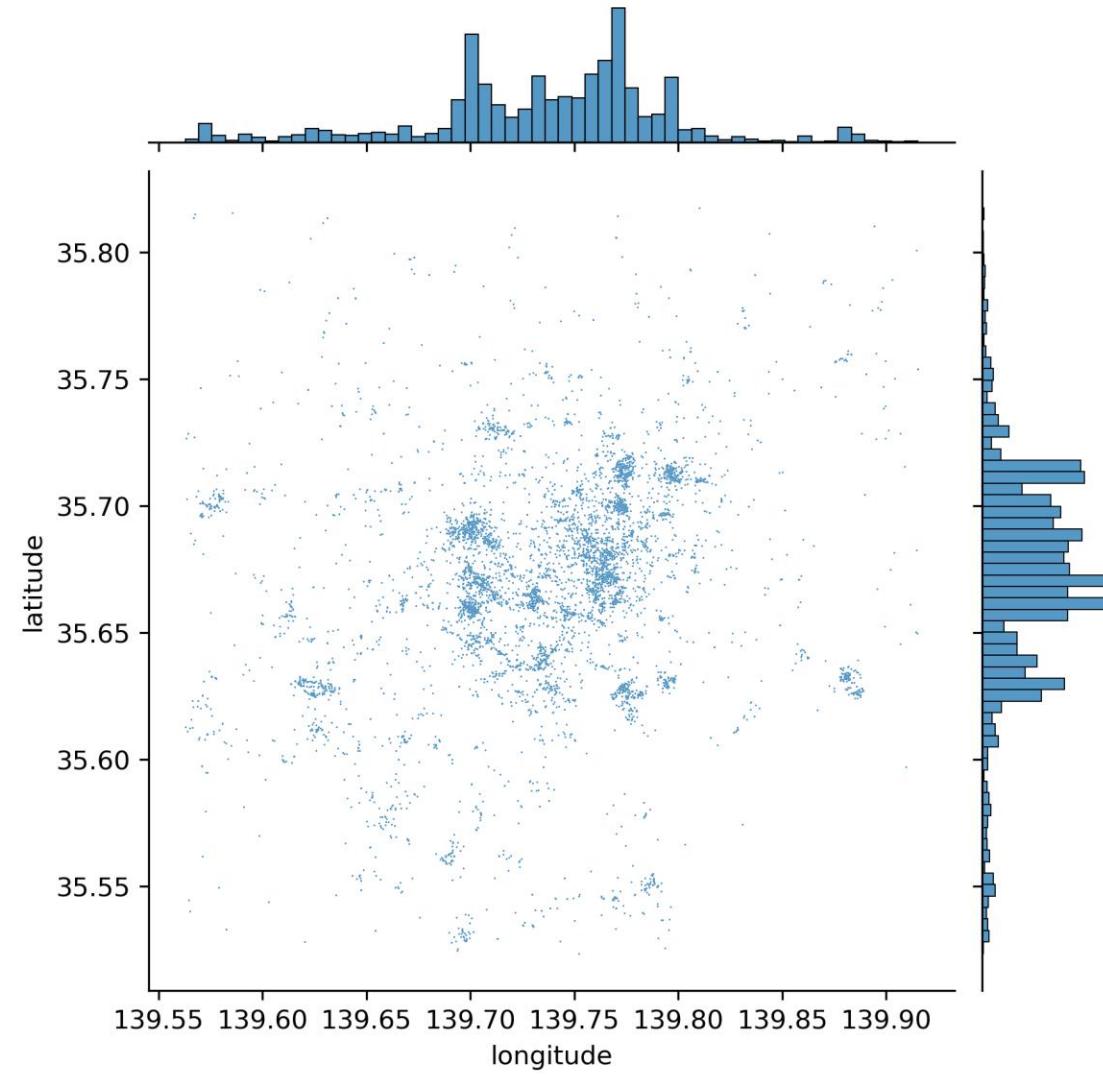


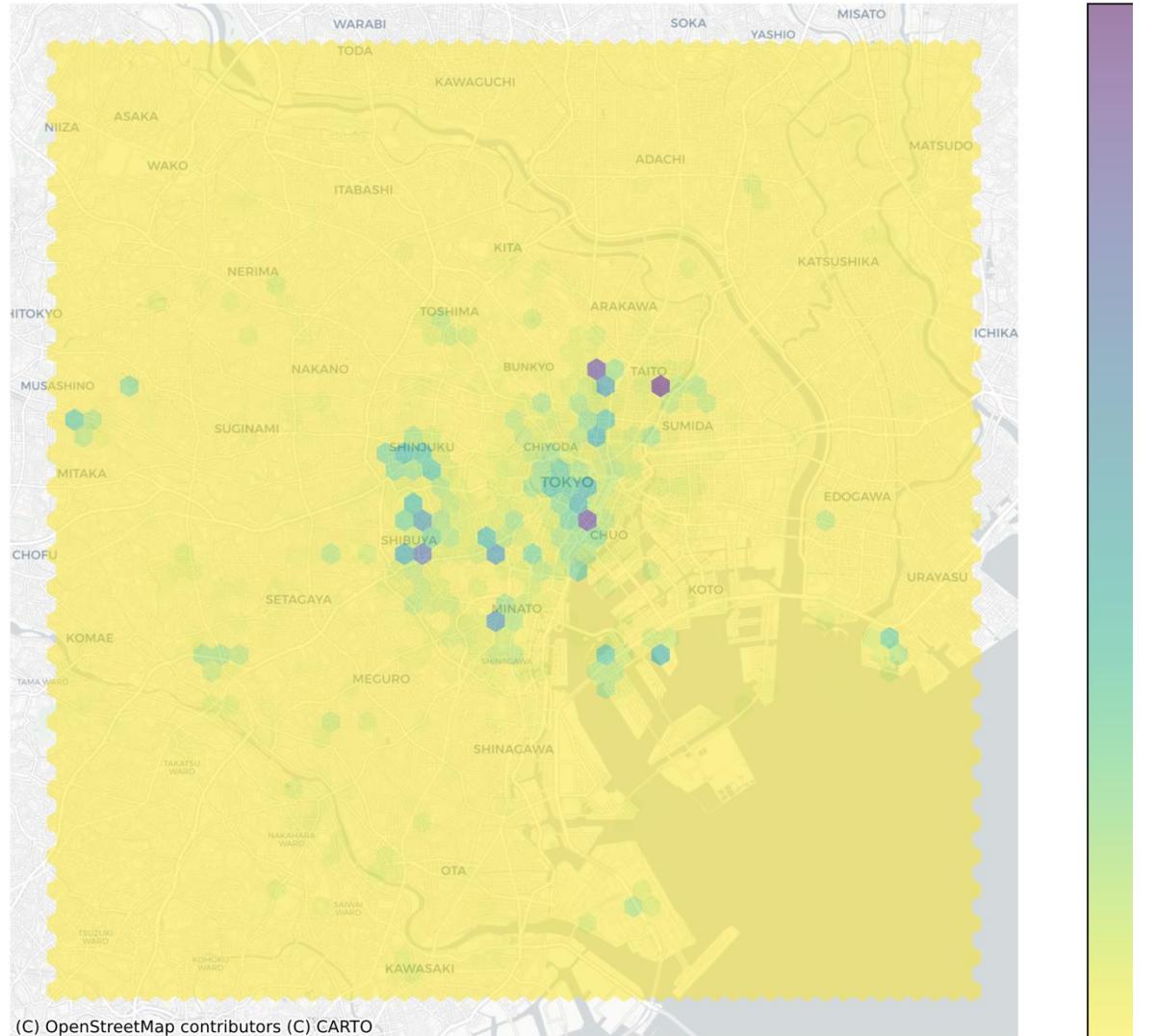
The challenge

- The pattern is only a reflection of the process
 - Use the pattern to uncover the process with limited visibility into patterns
-

EDA: Flickr photos in Japan

- Where do people take pictures?
- When are those pictures taken?
- Why do certain places attract many more photographers than others?





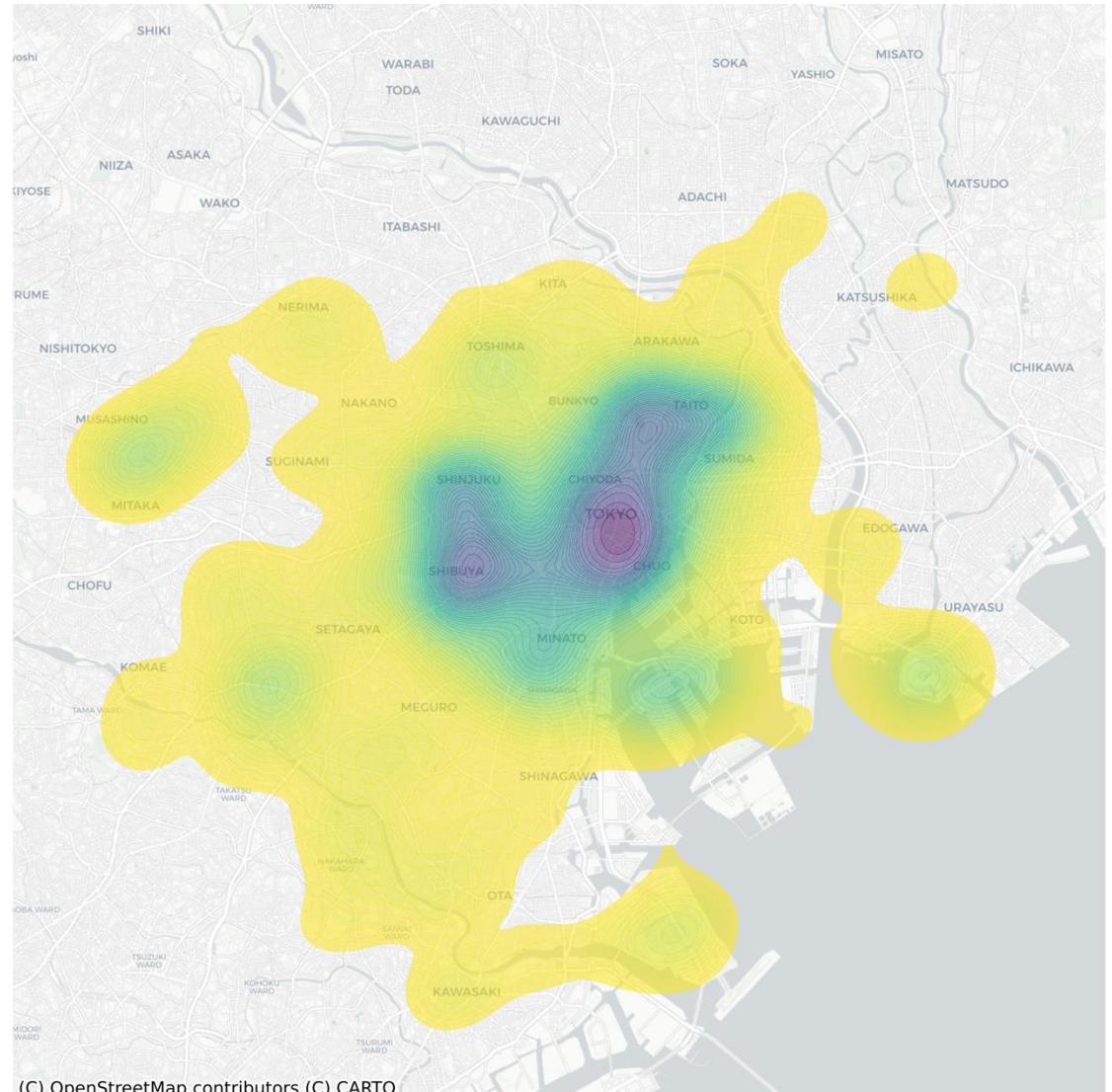
Analyze density

Hexbin

- More contiguous connectivity and uniform areas as compared to choropleths
- This is a **point to surface** transformation
 - Discretizes a continuous phenomenon

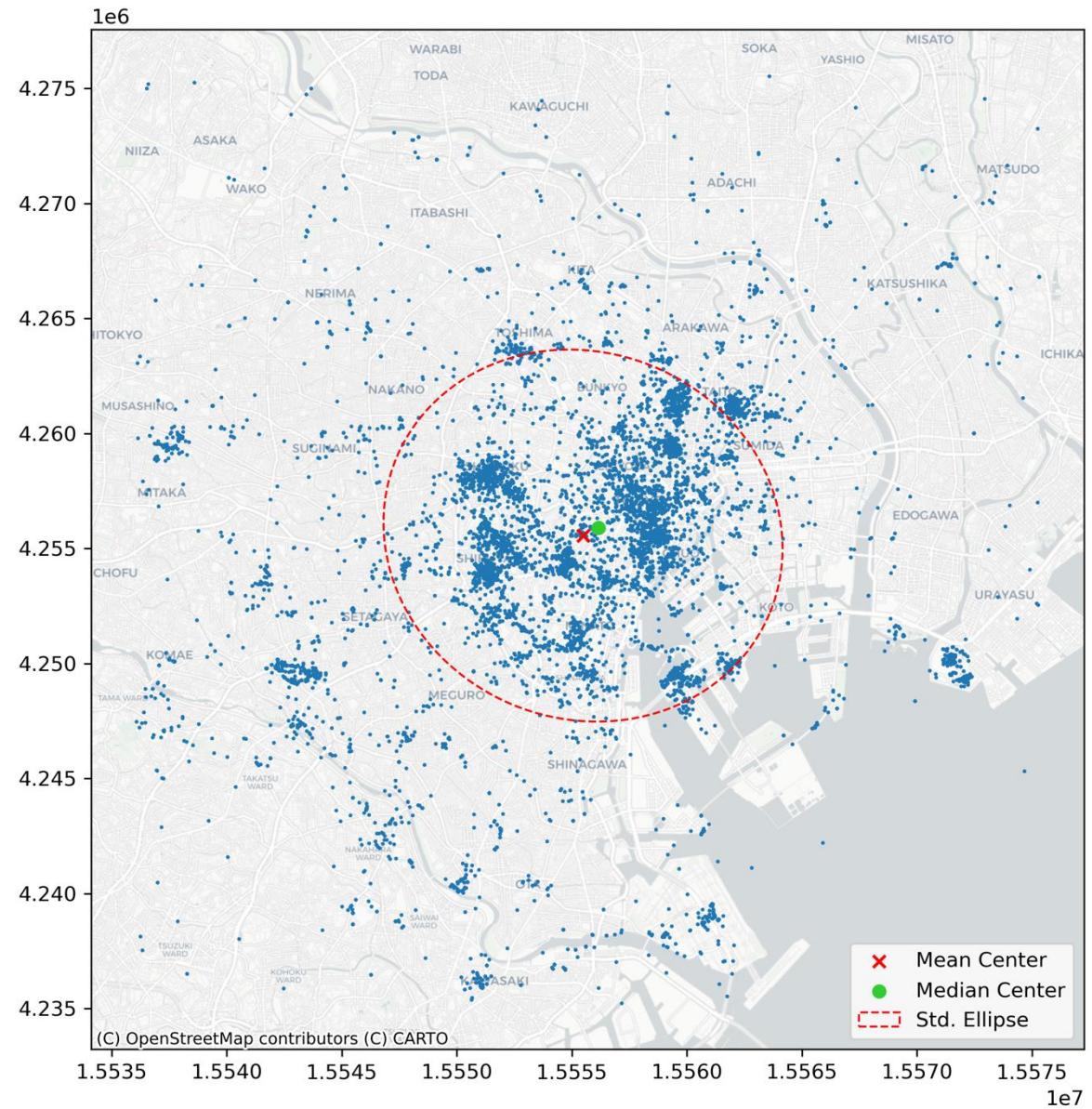
Kernel Density Estimation

- A KDE lays a grid of points over the space of interest on which it places kernel functions that count points around them with a different weight based on the distance.
- These counts are then aggregated to generate a global surface with probability
- Common kernel function – gaussian
- Plot using seaborn.kdeplot



Centrography

- Analysis of the location and dispersion of a phenomenon
 - Centrography.mean_center
 - Centrography.Euclidian_mean
 - Centrography.ellipse



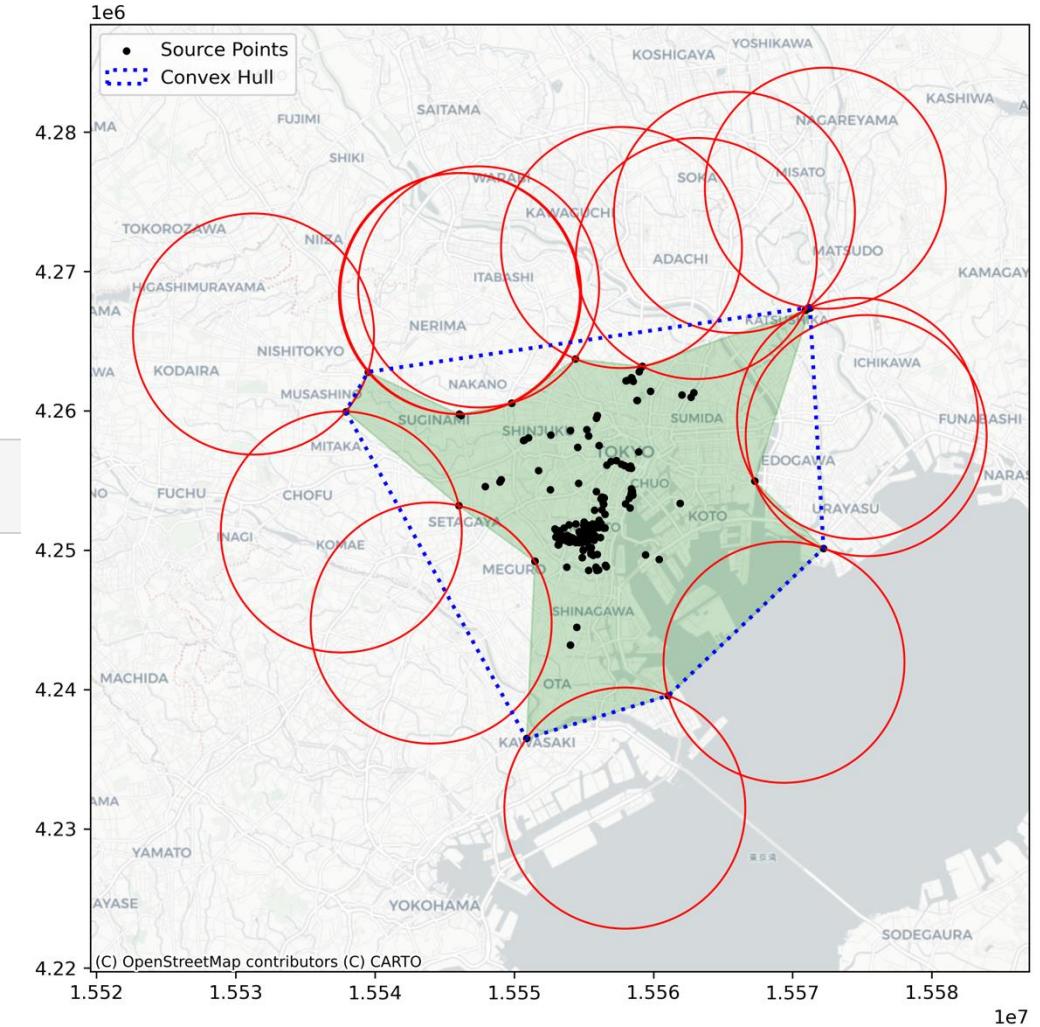
Measures from centrography

```
centrography.std_distance(db[["x", "y"]])
```

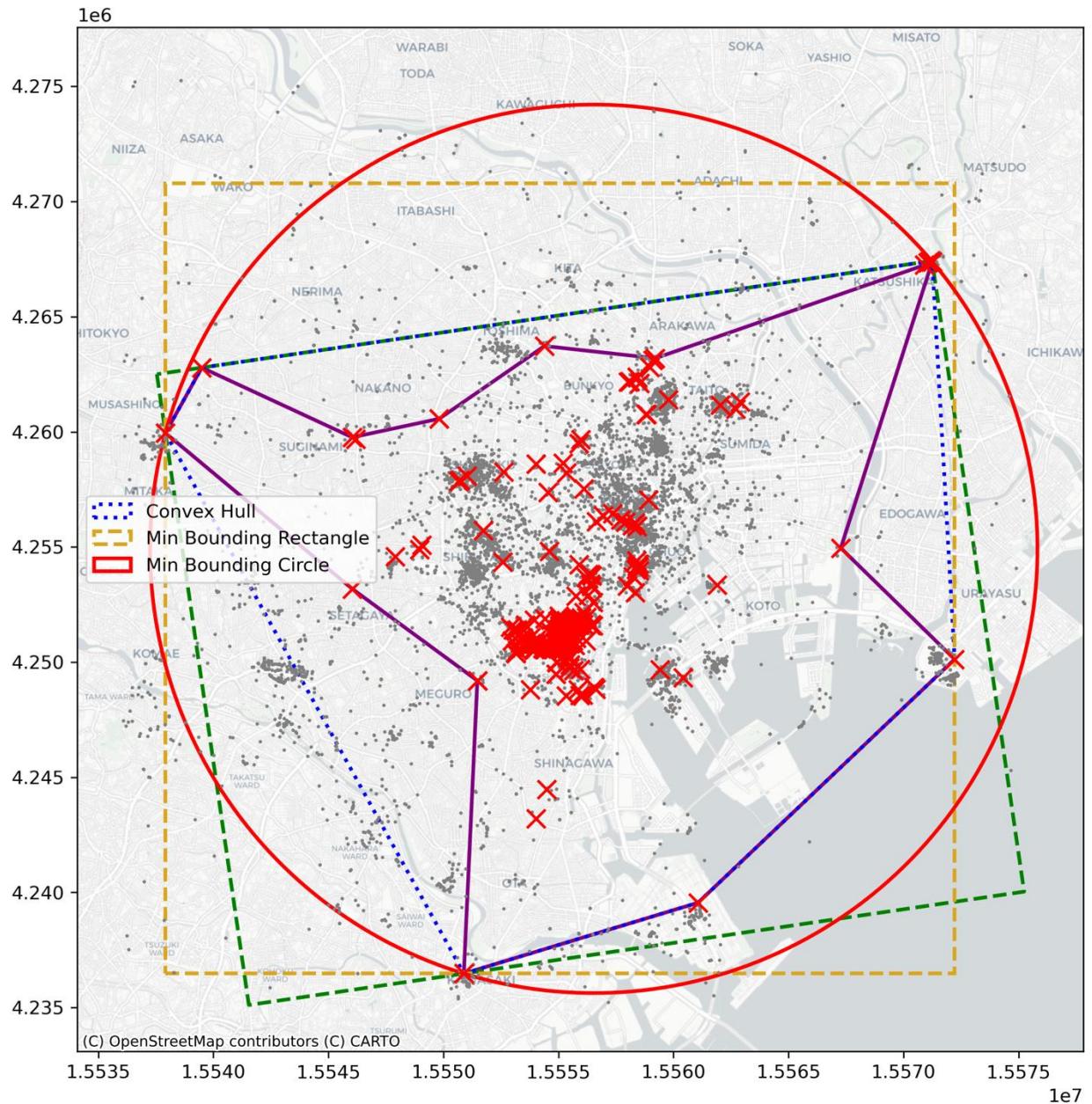
8778.218564382098

```
convex_hull_vertices = centrography.hull(coordinates)
```

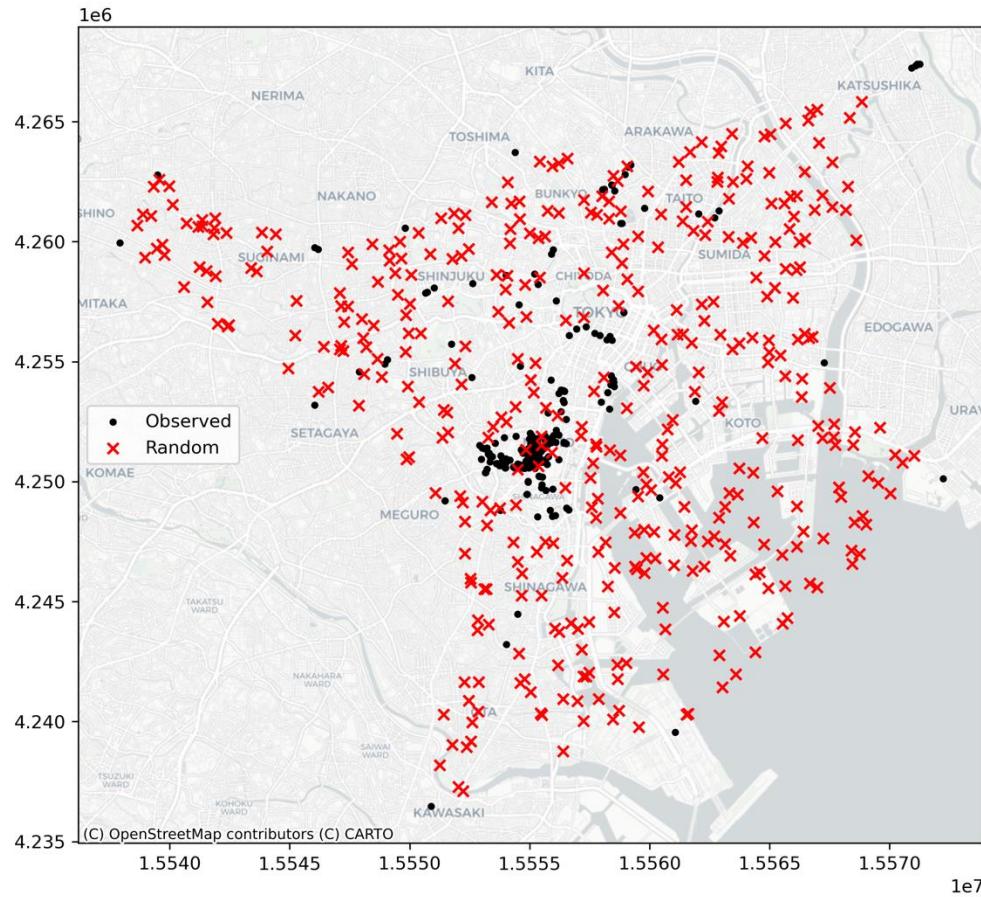
```
alpha_shape, alpha, circs = libpysal.cg.alpha_shape_auto(  
    coordinates, return_circles=True  
)
```



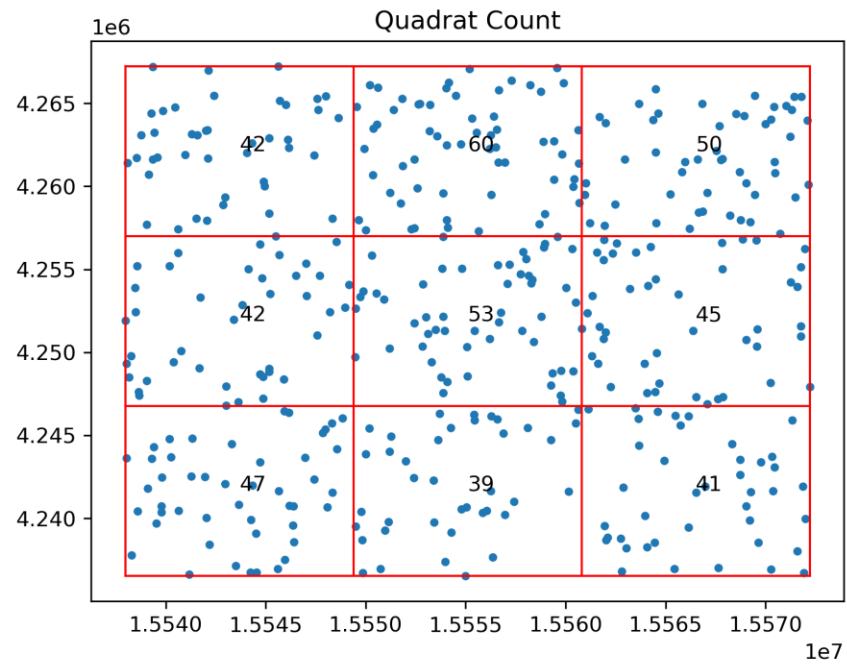
More bounds



Comparison with random data

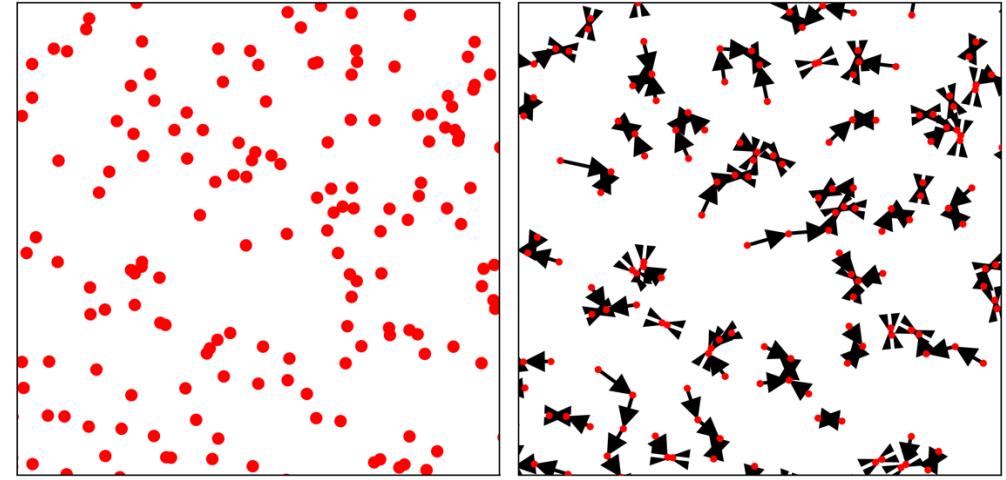


```
random_pattern_ashape = random.poisson(  
    alpha_shape, size=len(coordinates)  
)
```

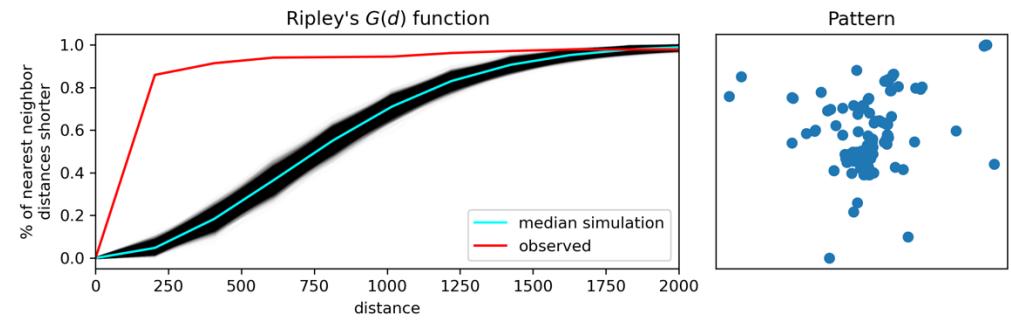


Ripley's G

- Finding distribution of nearest neighbors
- Rapidly increasing distance
 - Clustered points
- Slowly increasing distance
 - Dispersed points



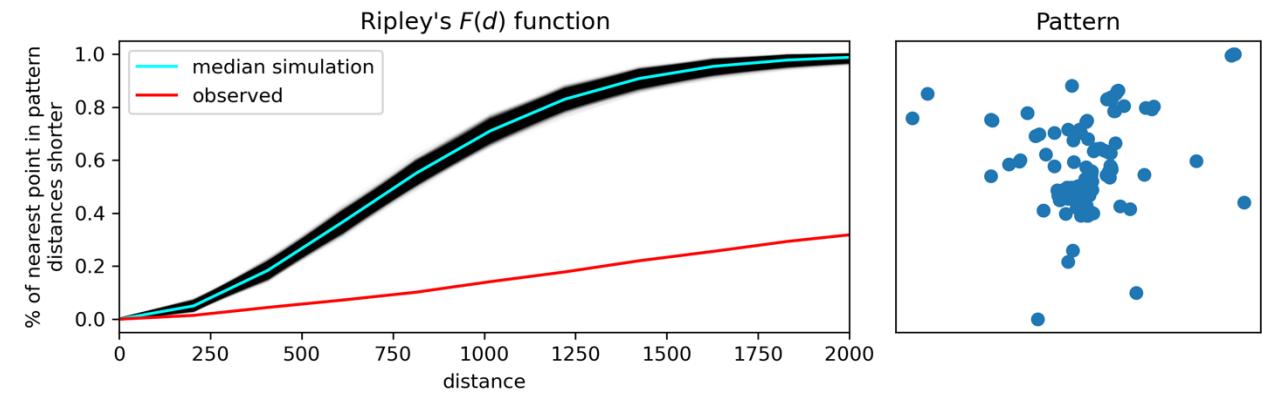
• Points
← Nearest Neighbor to Point



The rapid rise in observed shows a clustered pattern

Ripley's F

- Distance from random points in empty space to a point in the cluster
- A measure of dispersion for the data



Identifying cluster centers

DBSCAN : Density Based
Spatial Clustering of
Applications

Classify points into:

- *Noise*, for those points outside a cluster.
- *Cores*, for those points inside a cluster with at least m points in the cluster within distance r .
- *Borders*, for points inside a cluster with less than m other points in the cluster within distance r .

