Geographic Data Science

Visualisation of Point Patterns

Dani Arribas-Bel

Visualization of PPs

Three routes (today):

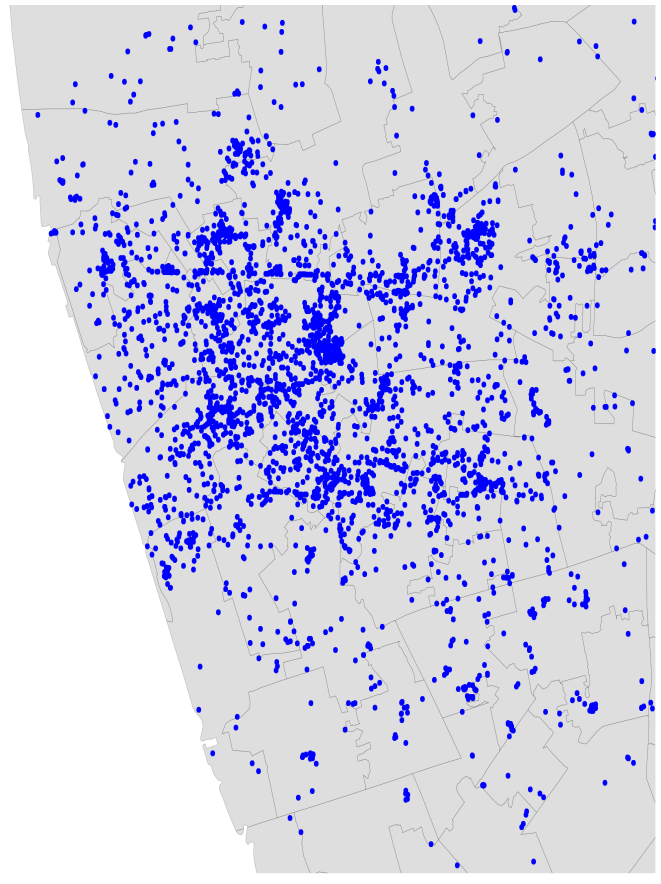
- *One-to-one* mapping ↔ "Scatter plot"
- *Aggregate* ↔ "Histogram"
- *Smooth* ↔ KDE

One-to-one

One-to-one

- Intuitive
- Effective in small datasets
- Limited as size increases until useless

One-to-one



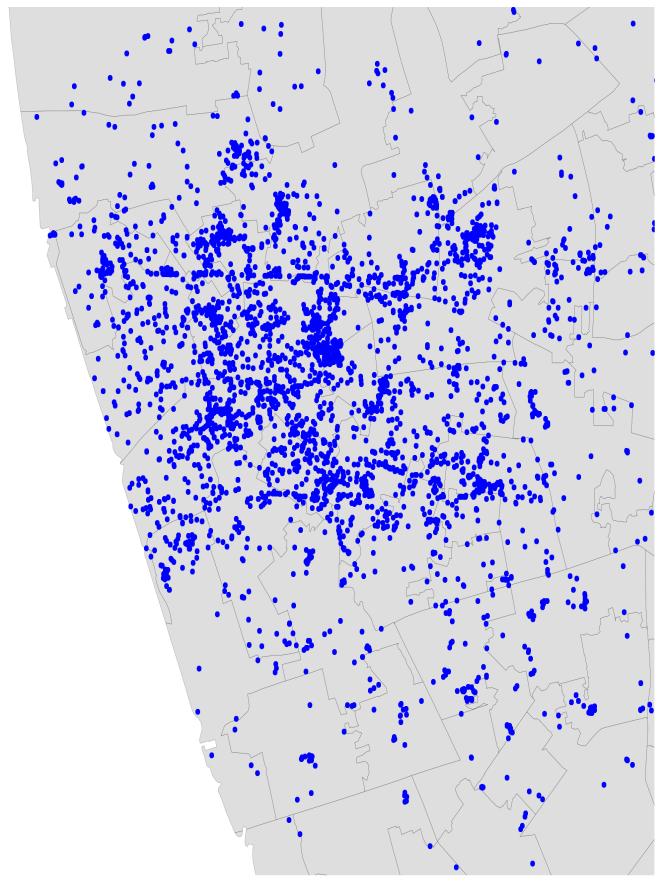
Aggregation

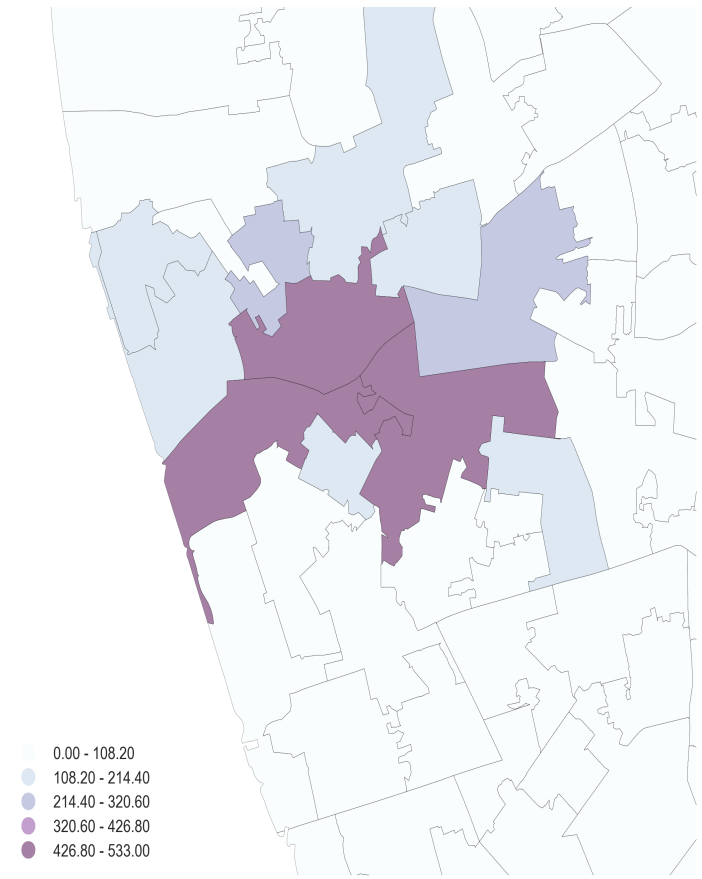
Points meet polygons

Use **polygon** boundaries and **count** points per area

[Insert your skills for **choropleth mapping** here!!!]

But, the polygons need to *"make sense"* (their delineation needs to relate to the point generating process)




0.00 - 108.20
108.20 - 214.40
214.40 - 320.60
320.60 - 426.80
426.80 - 533.00

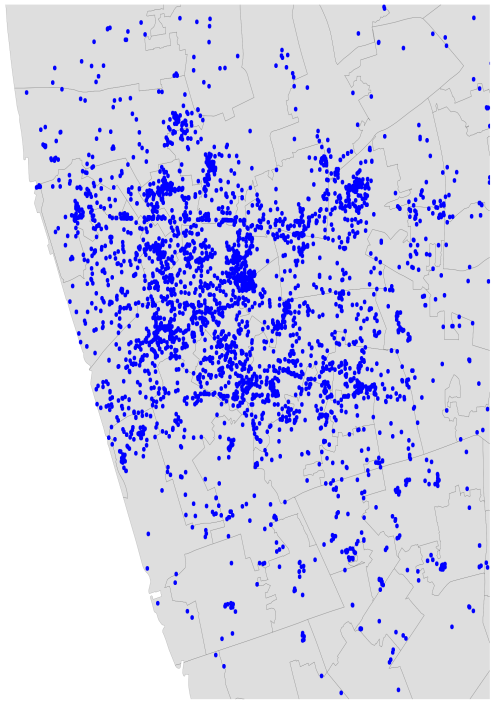
Hex-binning

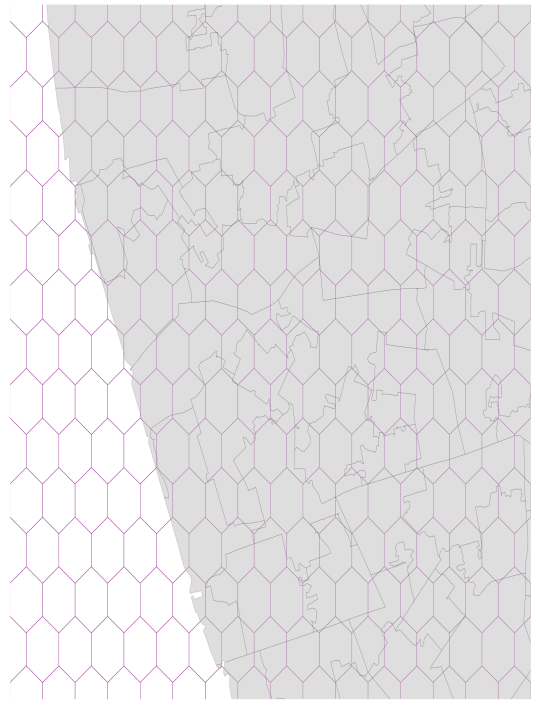
If **no** polygon boundary seems like a **good candidate** for aggregation…

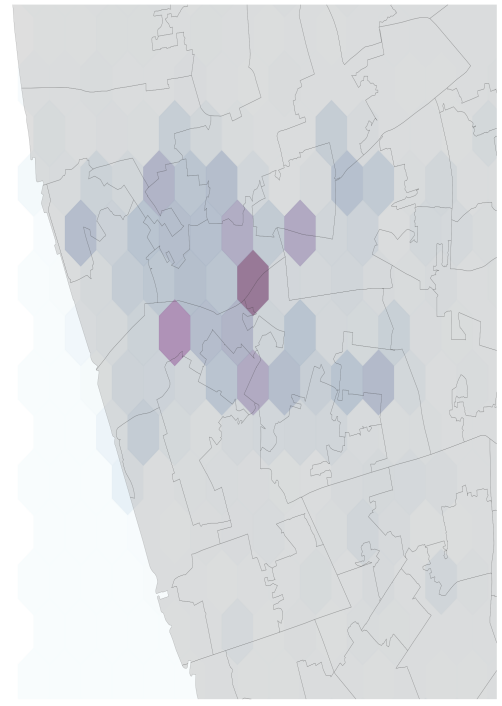
…draw a **hexagonal** (or squared) **tesselation**!!!

Hexagons…

- Are **regular**
- **Exhaust** the space (Unlike circles)
- Have **many sides** (minimize boundary problems)







But…

(Arbitrary) aggregation may induce **MAUP** (see Block D)

+

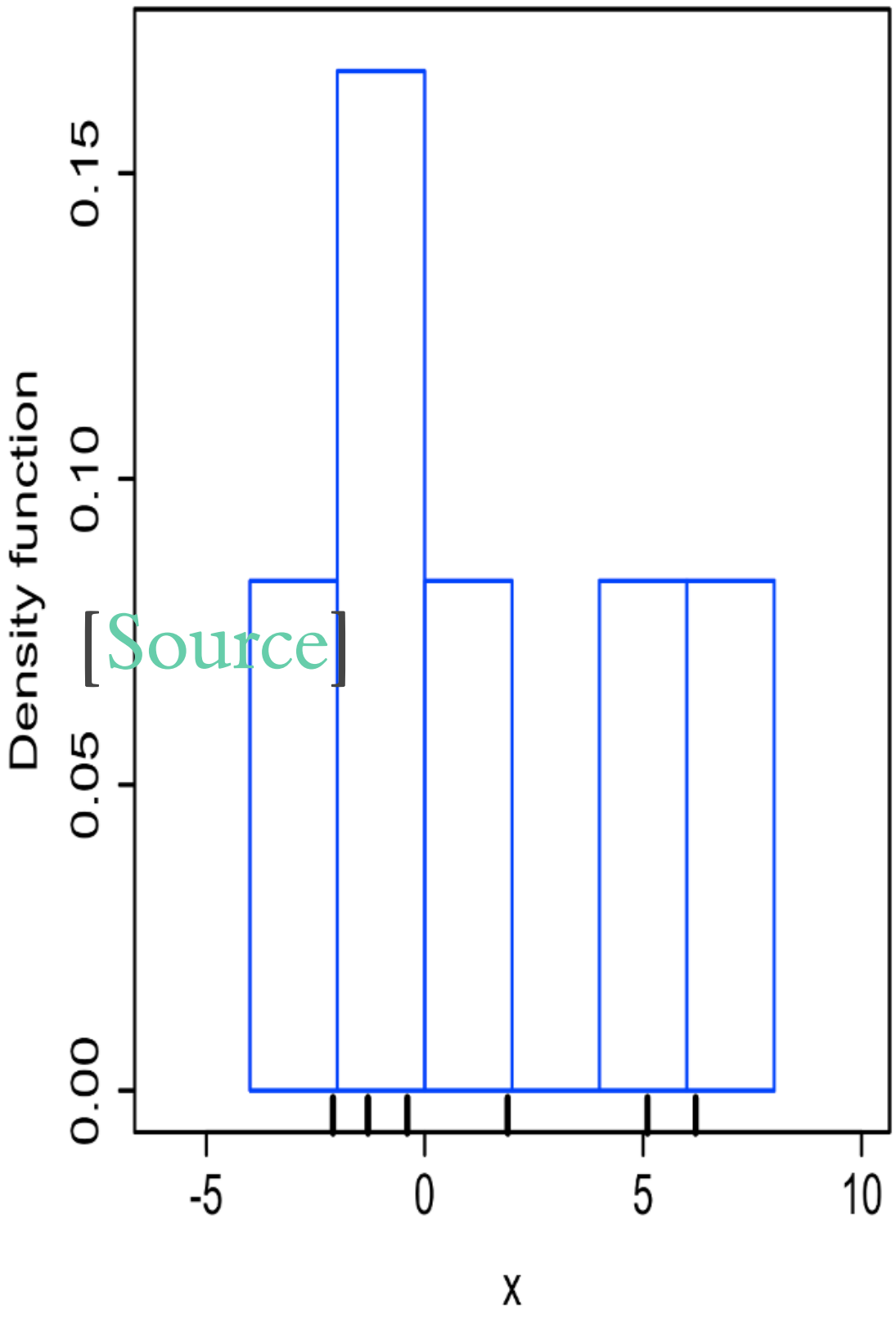
Points usually represent events that affect only **part** of the population and hence are best considered as **rates**

Kernel Density Estimation

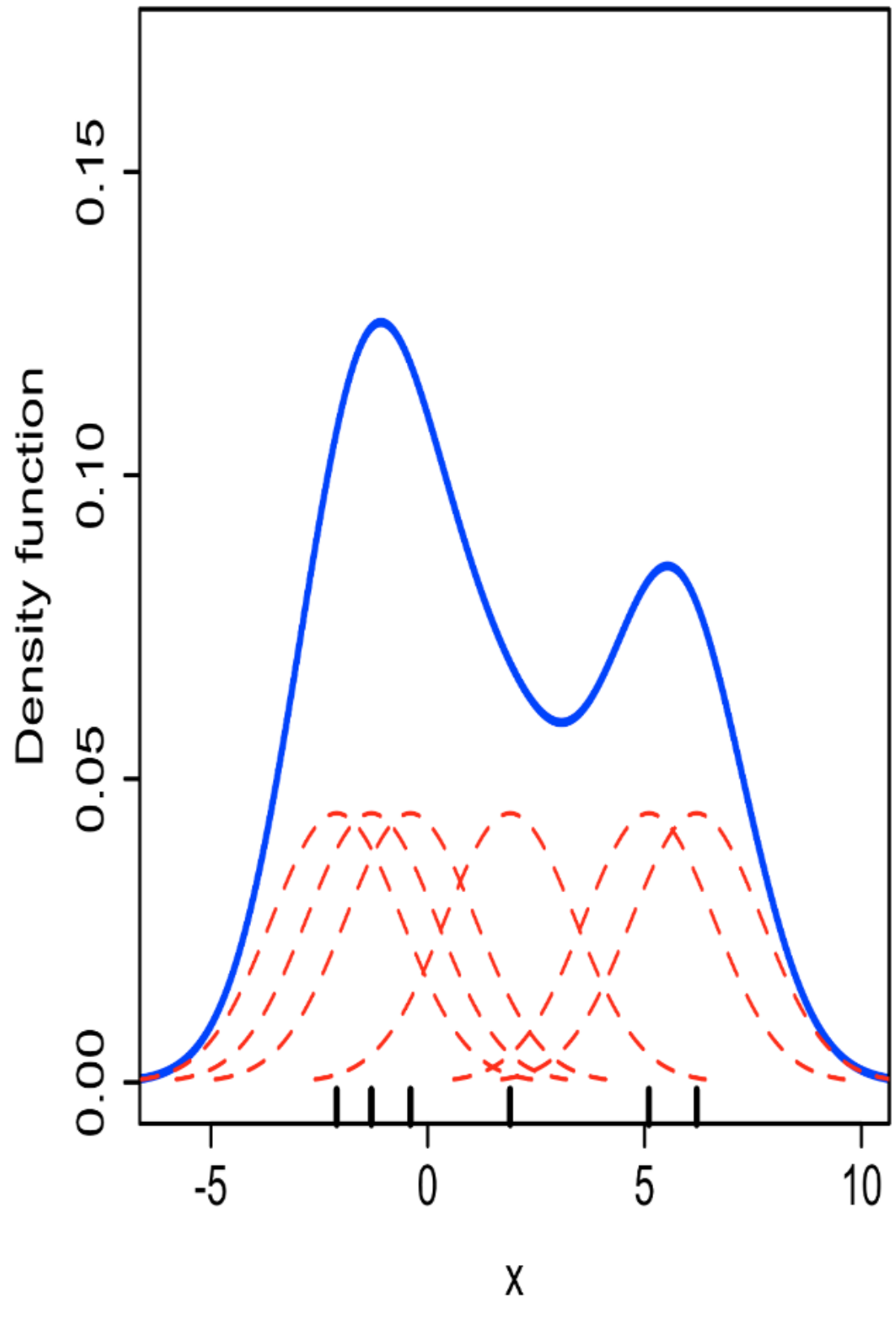
Kernel Density Estimation

*Estimate the (**continuous**) observed **distribution** of a variable*

- Probability of finding an observation at a given point
- "Continuous histogram"
- Solves (much of) the MAUP problem, but not the underlying population issue


Density function
0.00
0.05
0.10
0.15
-5
0
5
10
x

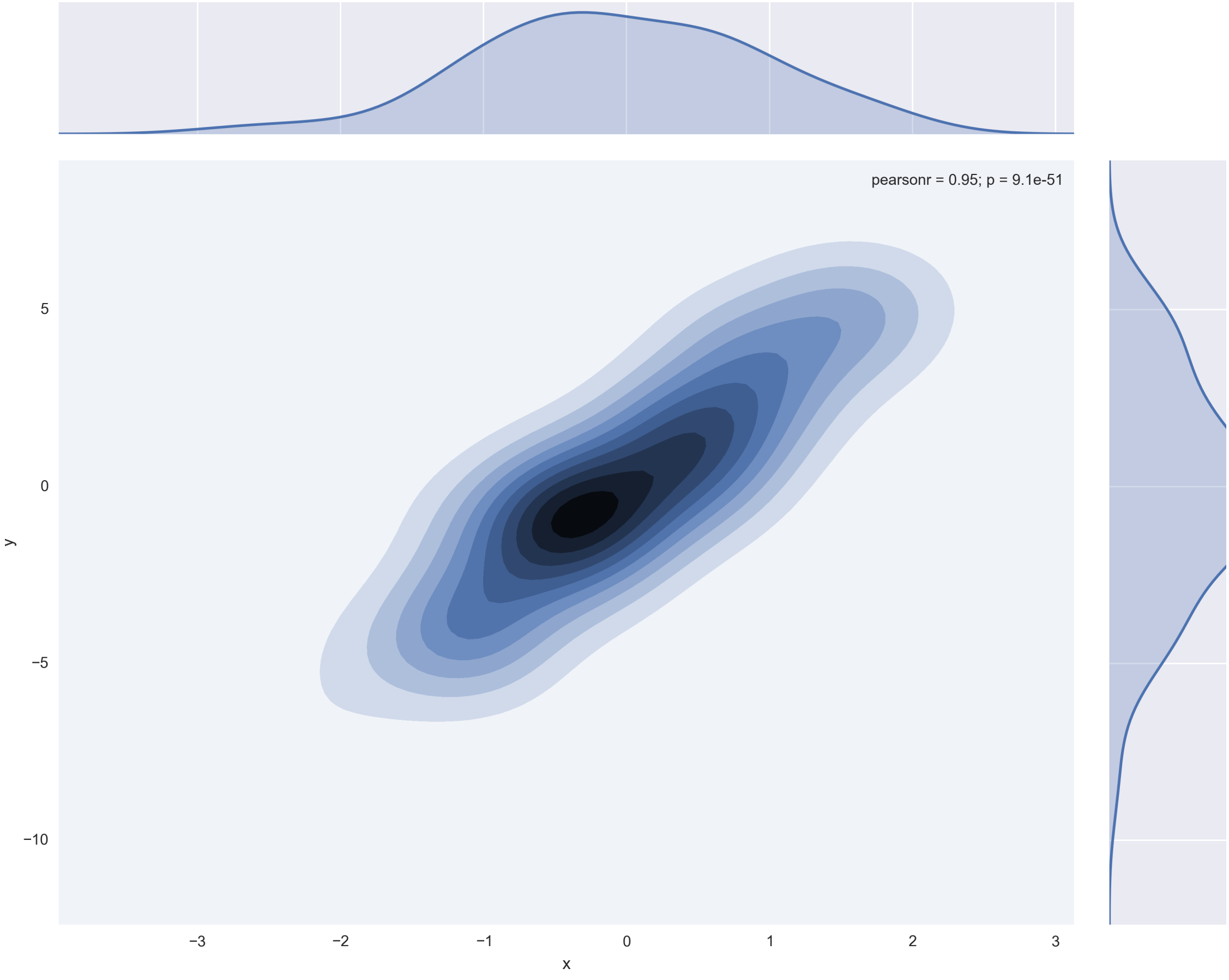
[Source]


Density function
0.00
0.05
0.10
0.15
-5
0
5
10
x

Bivariate (spatial) KDE

Probability of finding observations at a given point in space

- **Bivariate** version: distribution of **pairs of values**
- In **space**: values are coordinates (XY), locations
- Continuous "version" of a choropleth


pearsonr = 0.95; p = 9.1e-51
5
0
−5
−10
y
−3
−2
−1
0
1
2
3
x