Development of a random sampling tool for linear vector features using open-source GIS for Python

Trey Stafford

Senior in Geography, KU

11/20/2013

What does it really mean?

- Moving a set of lines around randomly!
- Taking advantage of existing GIS resources that are freely available.
- Utilization of a programming language to perform an otherwise tedious task.

What's the use?

- Random sampling of spatial datasets.
- Determining the location of sampling transects for field research.
- Other, unanticipated uses!

Why Python, and why open source?

- Python is well supported within the community and by commercial GIS providers.
- Potentially valuable contribution to GIS users. Should be accessible to a wide audience!
- Free and open source projects can be modified and redistributed as desired.
- It’s easy!

How does it work?

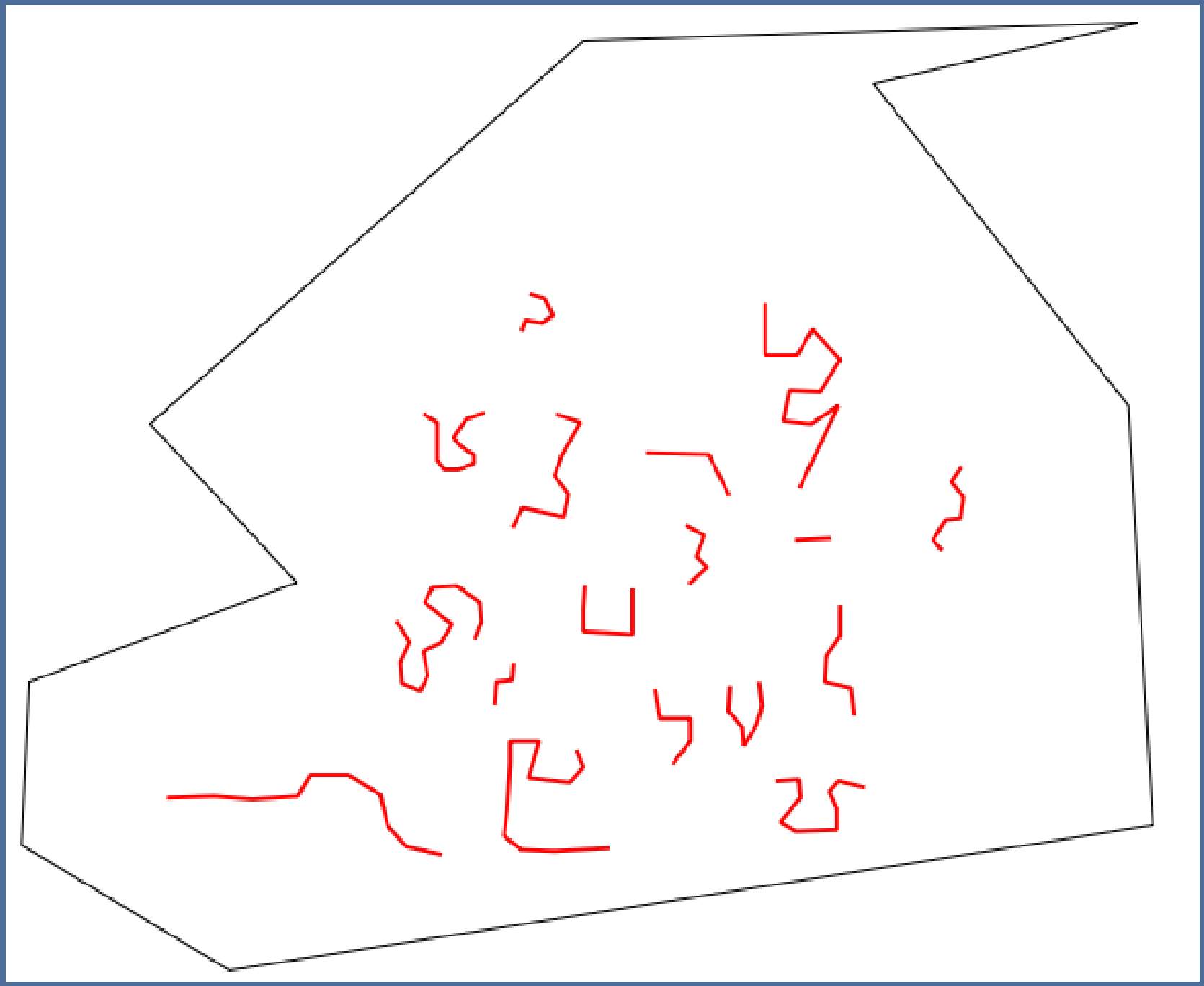
- User inputs: Shapefiles of linestring(s) and polygon + other parameters
- For each line in linestring(s):
  - A random point location within the polygon is selected.
  - X/Y offsets are determined and the line is translated to the random point.
  - Is the line sufficiently within the polygon at its new location?
    - If yes: save new location.
    - If no: try a new point.
- Save new locations of all linestrings in an output Shapefile.

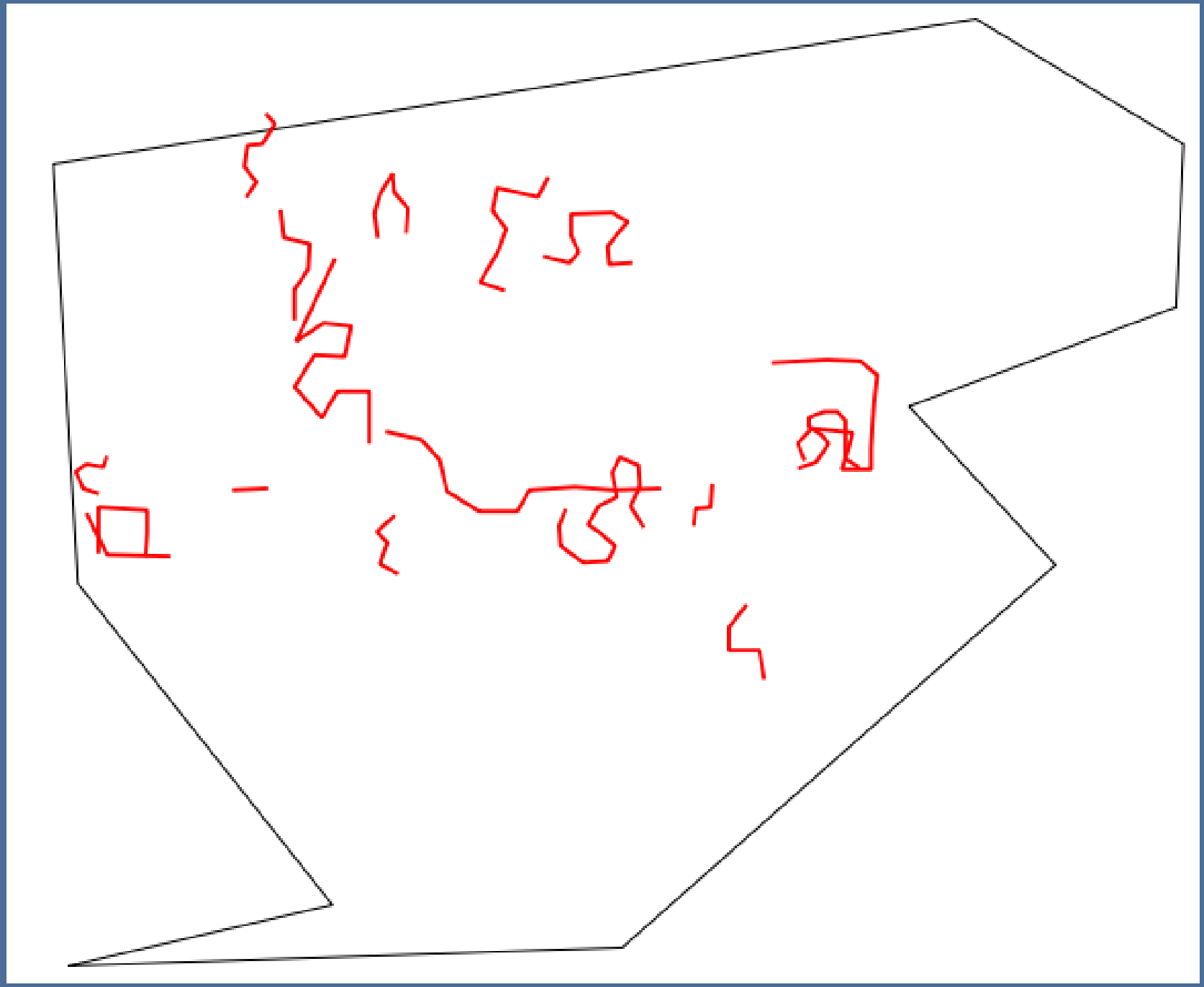
Details

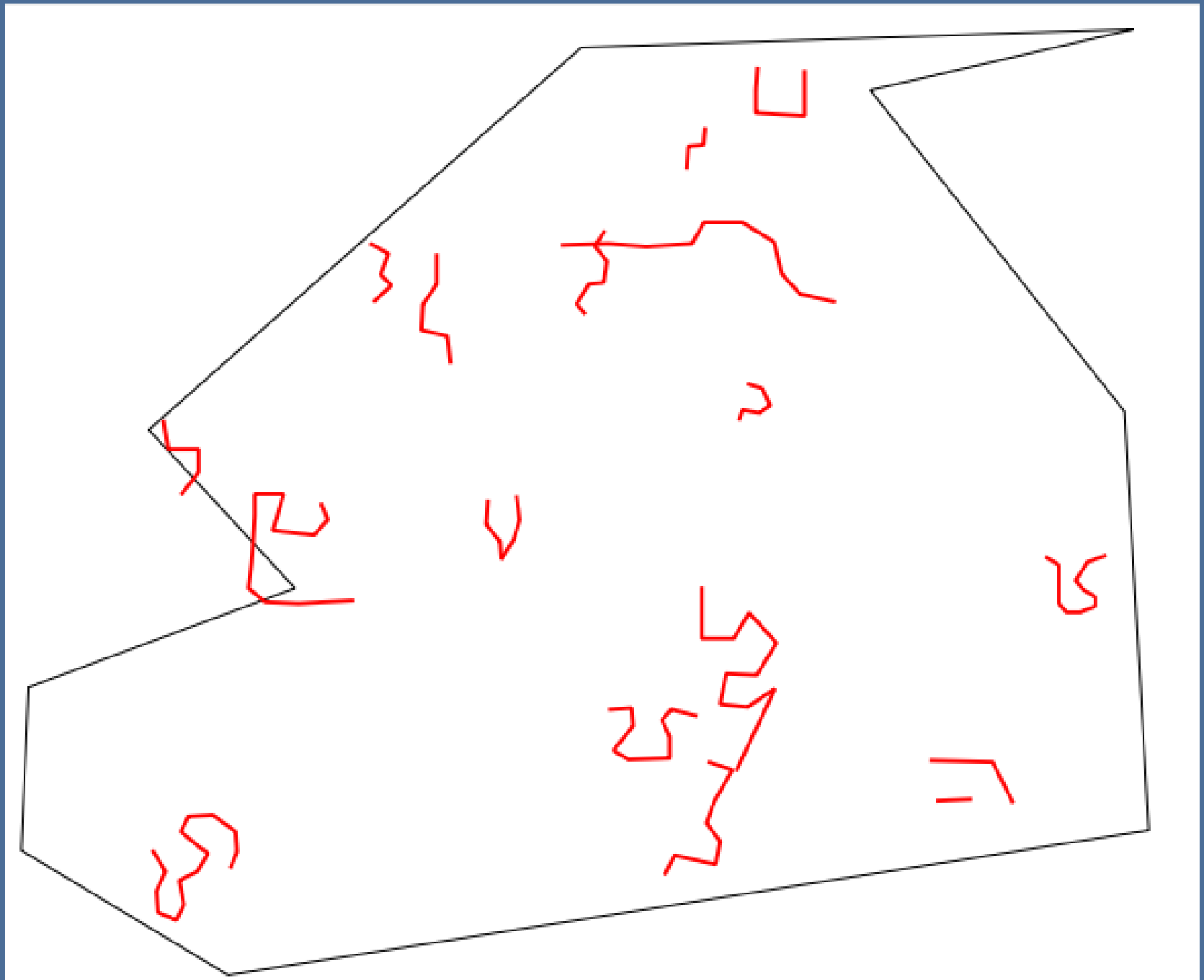
- Fiona + Shapely = Geometry reading/writing + manipulation.
- Other parameters:
  - How the line is translated to the random point.
  - How much of the line needs to be within the polygon for it to be considered “sufficiently within”?

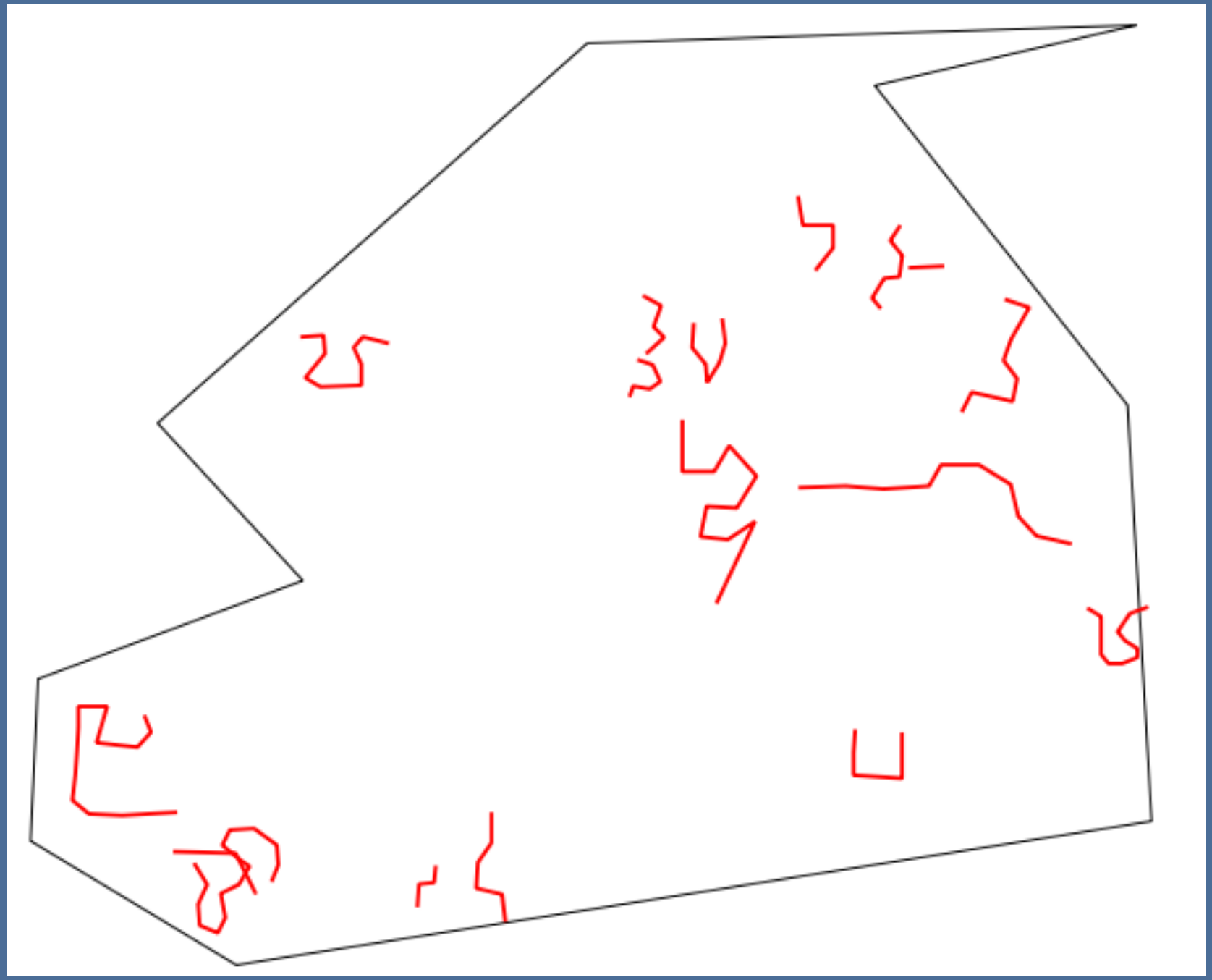
A Hypothetical Example

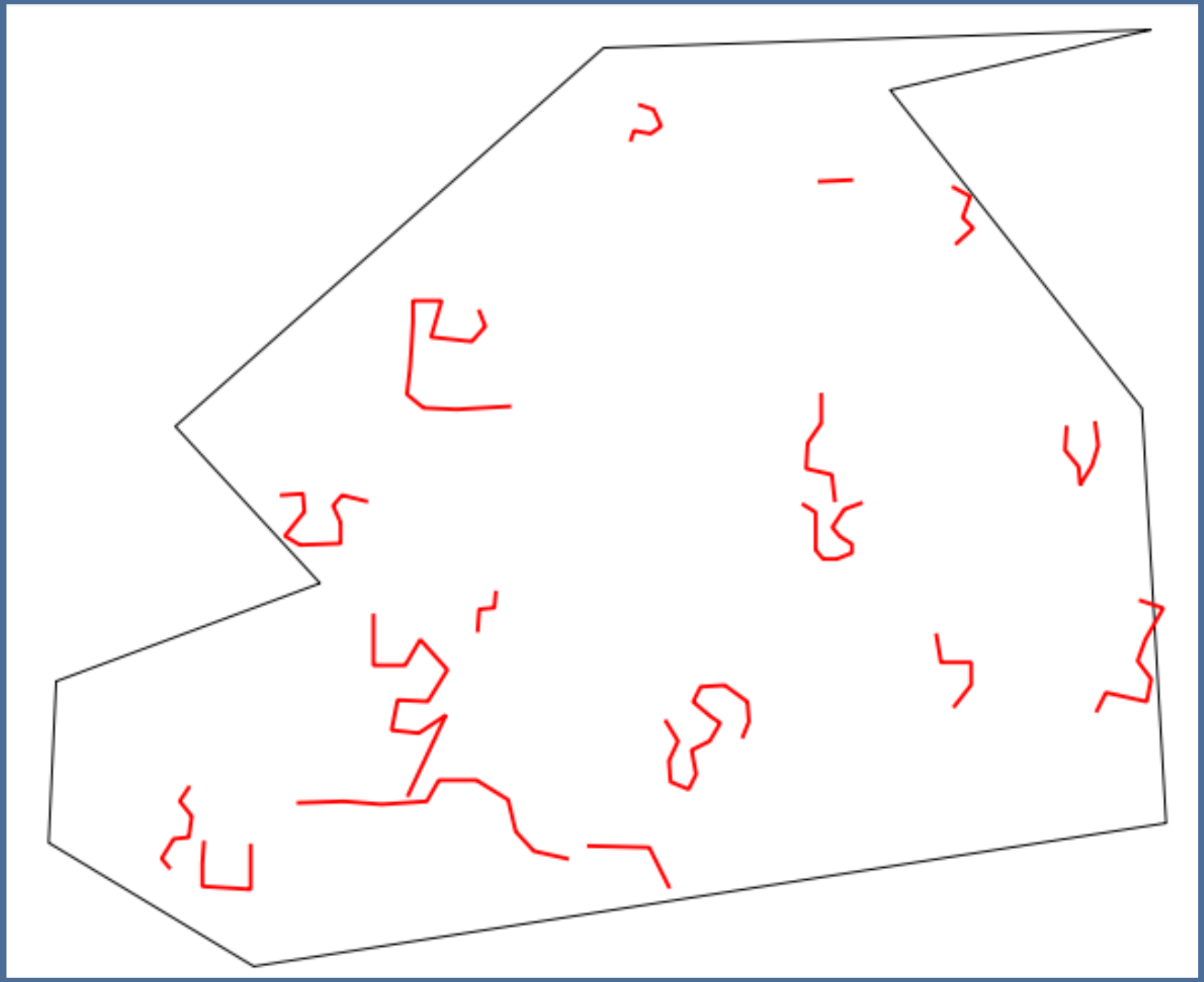
- Number of lines: 17
- Line origin set to ‘all’ with a point count of 1.
  - Tries to move line to random point at every vertex until the line is sufficiently in the polygon.
  - If none of the line vertices work, generate new random point.
- Threshold set to 80%
  - Defines what ‘sufficiently’ means.
  - 80% of line must lie within polygon.





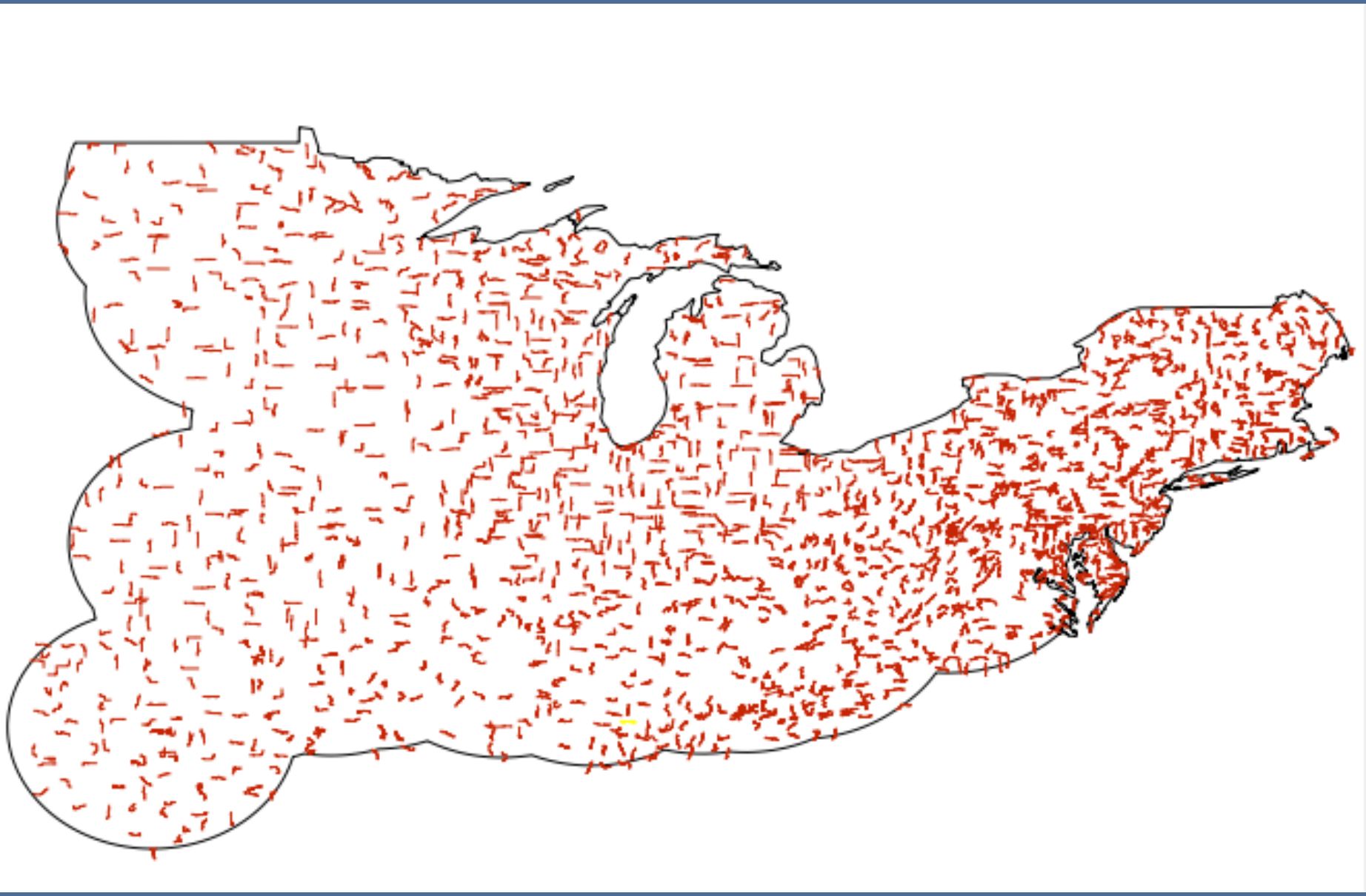


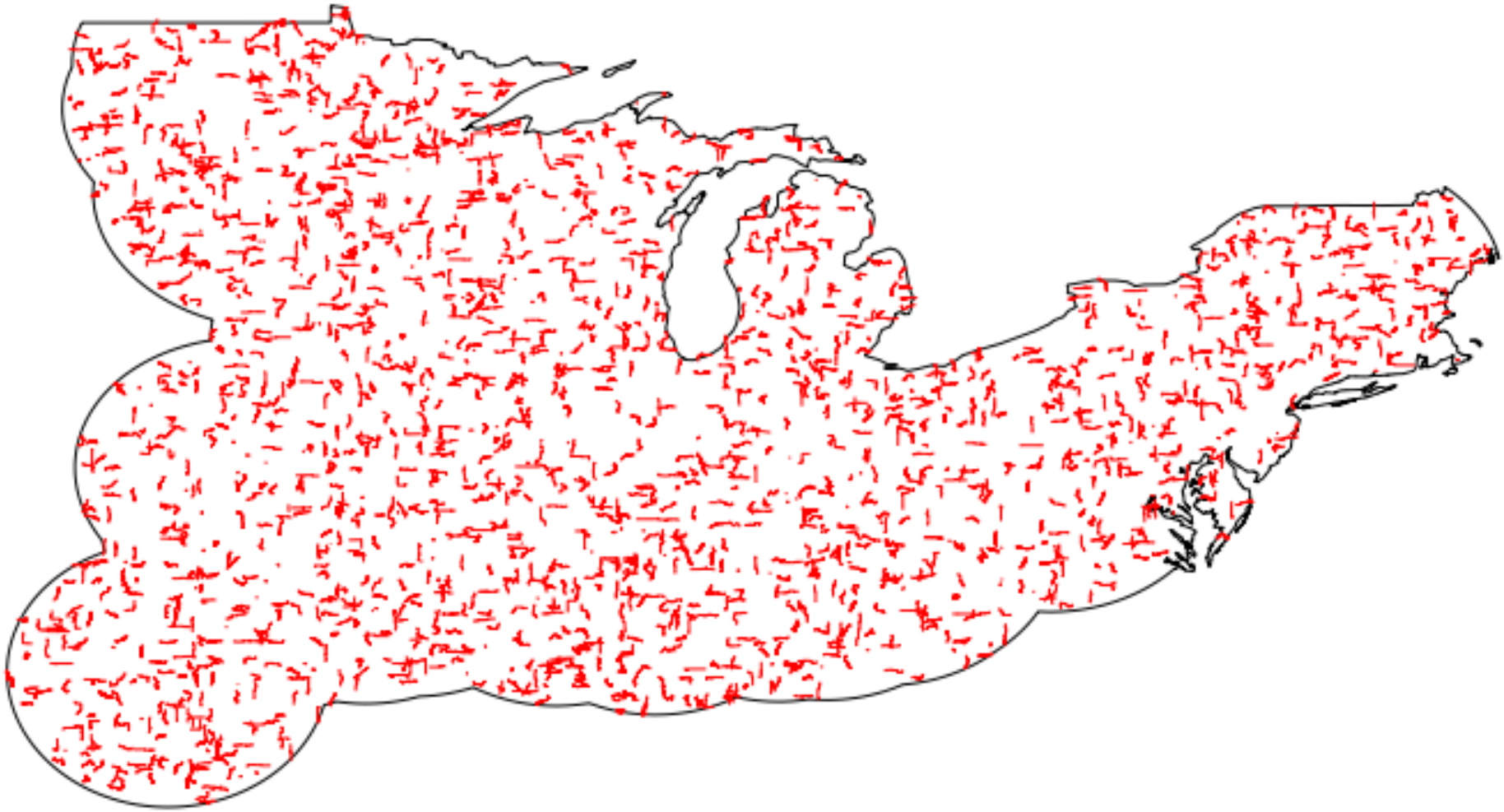




A Real World Example

- Number of lines: 1997
- Line origin set to ‘all’ with a point count of 10
- Threshold set to 80%





The Future!

- Optimization and improvements to current implementation.
  - Parallel processing
  - Spatial indices
- Random rotation
- Creation and random placement of *new* lines.
- Polygon randomization

Acknowledgments

- Lynnette Dornak, PhD., University of Kansas.
- A. Townsend Peterson, Distinguished Professor, Ecology and Evolutionary Biology at the University of Kansas
- Sean Gillies and the other developers who created Shapely and Fiona.

Questions?