LASEK: LLM-Assisted Style Exploration Kit for Geospatial Data

Tarlan Bahadori, Sai Sreekar Sarvepalli, and Ahmed Eldawy, University of California, Riverside

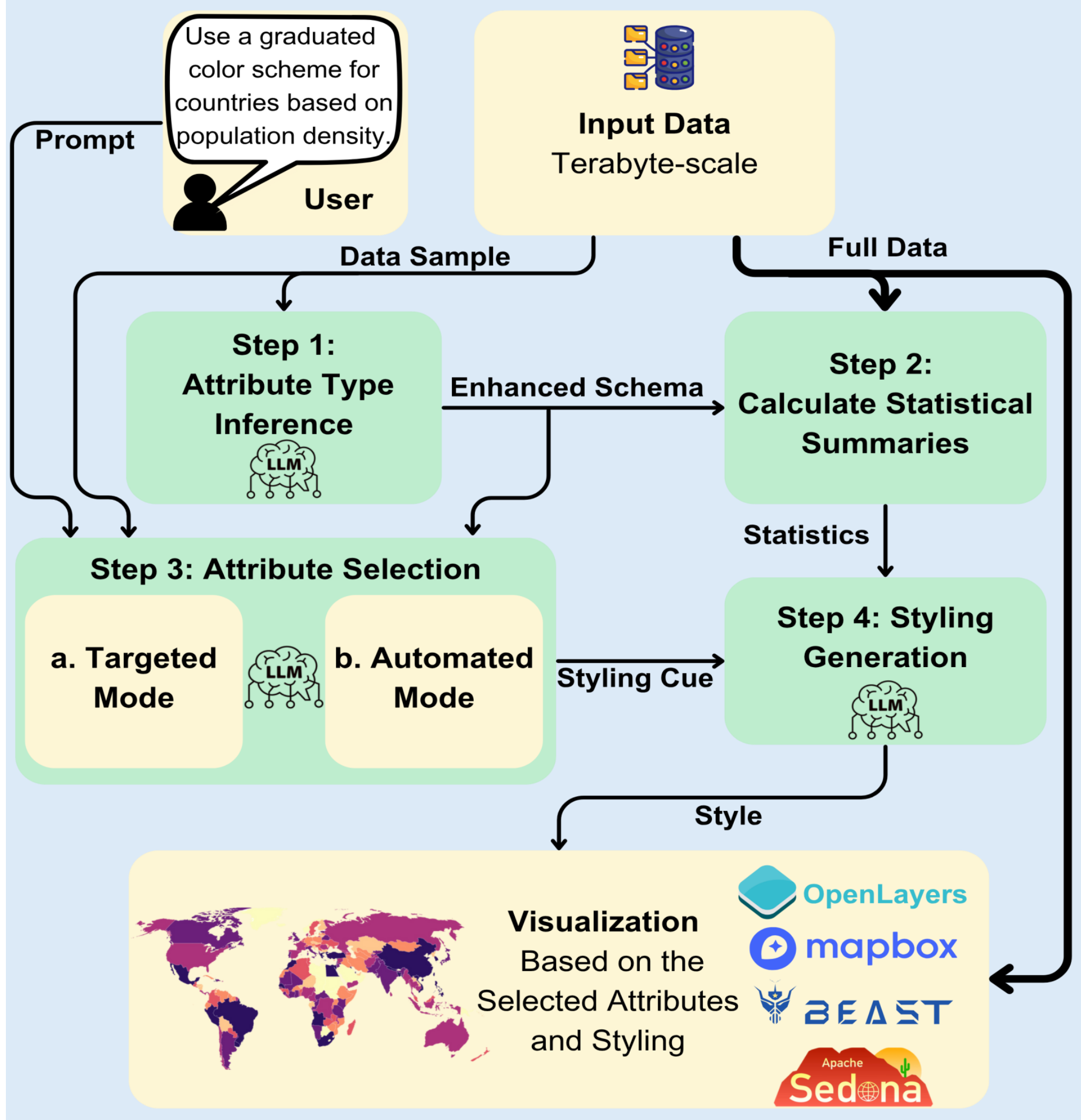
Background and Motivation

Data Complexity: Hundreds of unclear, undocumented fields require time-consuming manual exploration.

Flexibility vs. Scalability: Server-side tools limit customization; client-side tools struggle with big data.

Our goal: Combine scalability and interactivity for large-scale spatio-temporal visualization.

System Overview


Use a graduated color scheme for countries based on population density.
Prompt
User
Input Data
Terabyte-scale
Data Sample
Full Data
Step 1: Attribute Type Inference
LLM
Enhanced Schema
Step 2: Calculate Statistical Summaries
Statistics
Step 3: Attribute Selection
a. Targeted Mode
b. Automated Mode
Styling Cue
Step 4: Styling Generation
Style
Visualization Based on the Selected Attributes and Styling
OpenLayers
mapbox
BEAST
Apache Sedona

Targeted Mode: LLM generates the styling based on a specific prompt by user

Automated Mode: List of style suggestions for all attributes by LLM, user selects from the list

Visualization rendered **client-side**; styles cached per workspace and can be re-applied instantly

1) Enhanced schema: Data Sample + Schema is sent to LLM
LLM response: Detected types/units (Enhanced schema)
2) Statistical Summaries: Compute HLL distinct counts, top-k values, numeric min/max/avg/std, datetime min/max, and geometry MBR/types in one SparkSQL pass.
3) Styling Cue: Automated Mode: Stats sent to LLM, response: Style for all attributes
Targeted Mode: Stats + User Prompt sent to LLM, response: Specific style most suitable for user prompt
4) Style: LLM response (JSON) → JS code → Applied to the full data

Stats sent to LLM; LLM infers best style per attribute → UI shows suggested style

Examples:

{ **"continent":** "countDistinct": 6, "isDatetime": false, "topKValues": ["Africa", "Asia", "Europe",]}

→ Categorical attribute, 6 classes → suggest: categorized style + legend entries for each class. → UI: Color-by-category, one swatch per continent + legend.

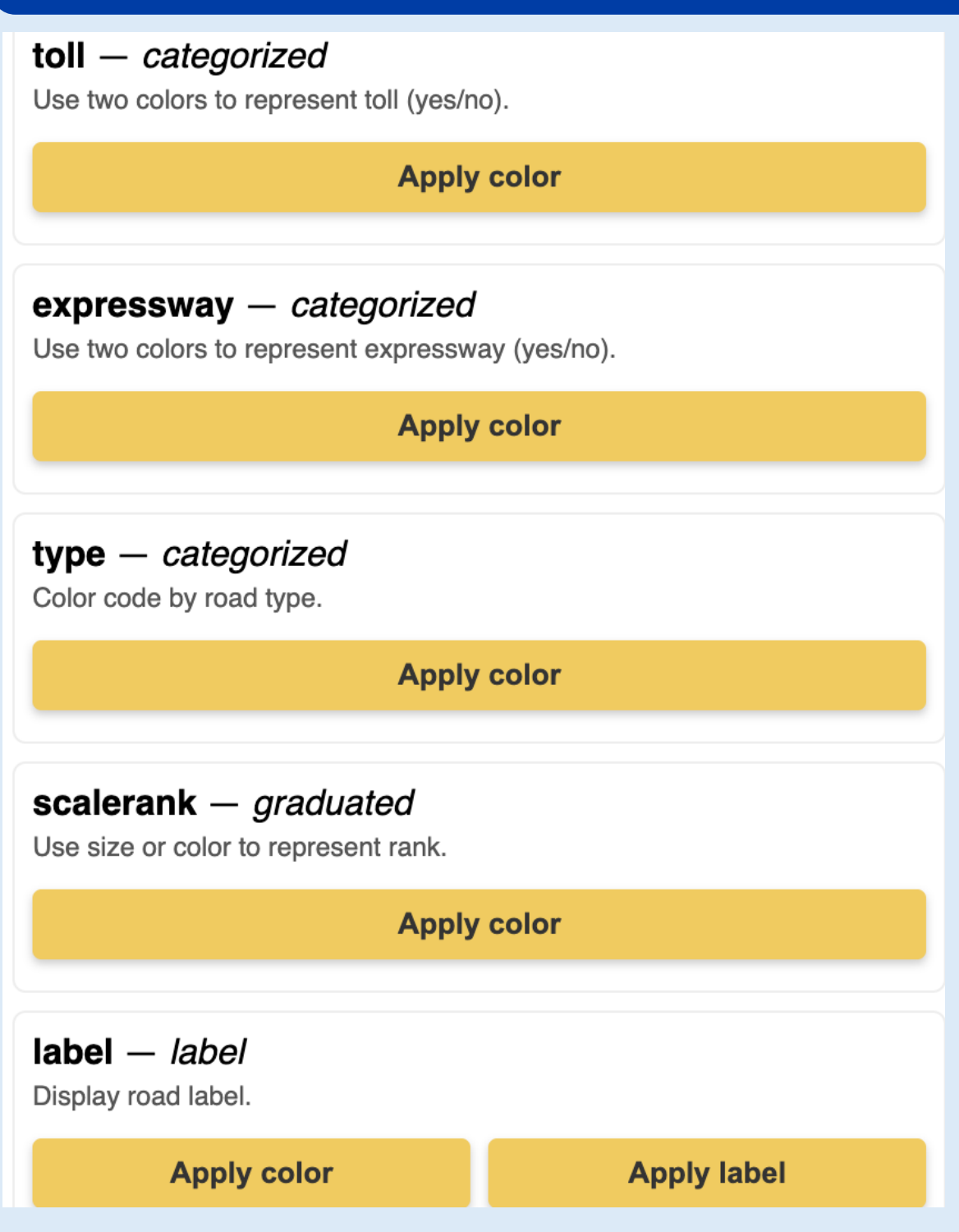
{**"pop_est":** "countDistinct": 181, "max": 1400000000, "mean": 36000000, "min": 1200, "stddev": 145000000}

→ Continuous, high variance → suggest: Graduated style; quantile bins; show value ranges in legend. → UI : 10-bin quantile ramp; legend shows value ranges

First prompt: Instructions + Schema + sample
Result: Detect attribute types, date type format, (Enhanced schema)

Second prompt: Instructions + Stats
Result: Attribute Style Suggestions + Explanations

Automated Mode:



toll — *categorized*
Use two colors to represent toll (yes/no).
Apply color

expressway — *categorized*
Use two colors to represent expressway (yes/no).
Apply color

type — *categorized*
Color code by road type.
Apply color

scalerank — *graduated*
Use size or color to represent rank.
Apply color

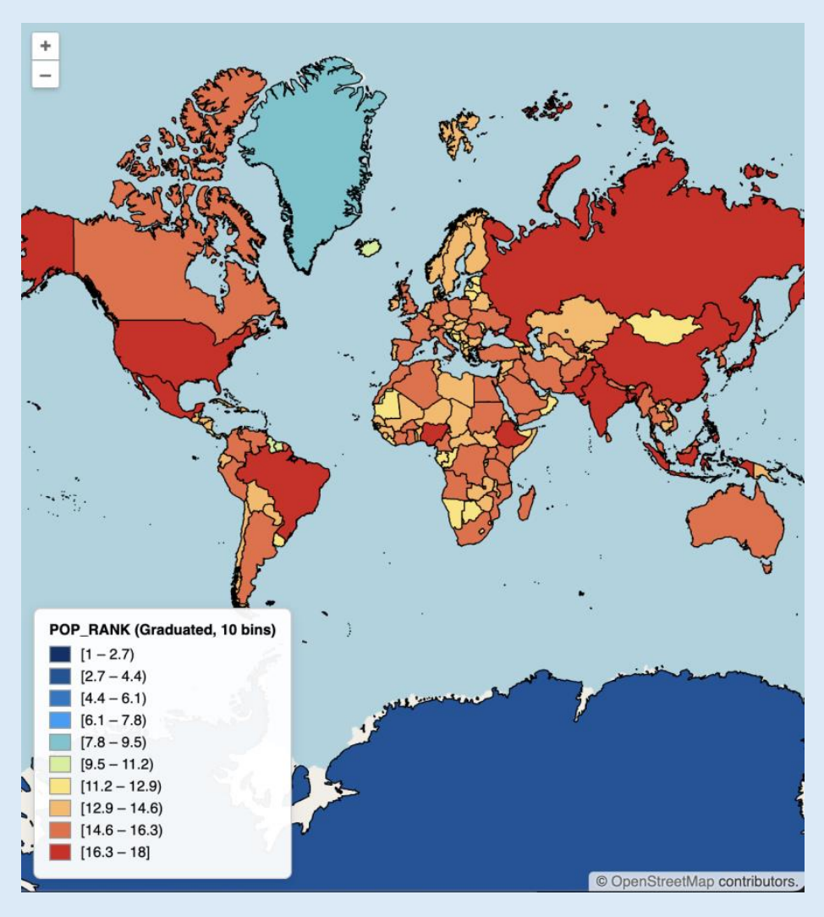
label — *label*
Display road label.
Apply color | Apply label

UI and Interaction

Targeted Mode:

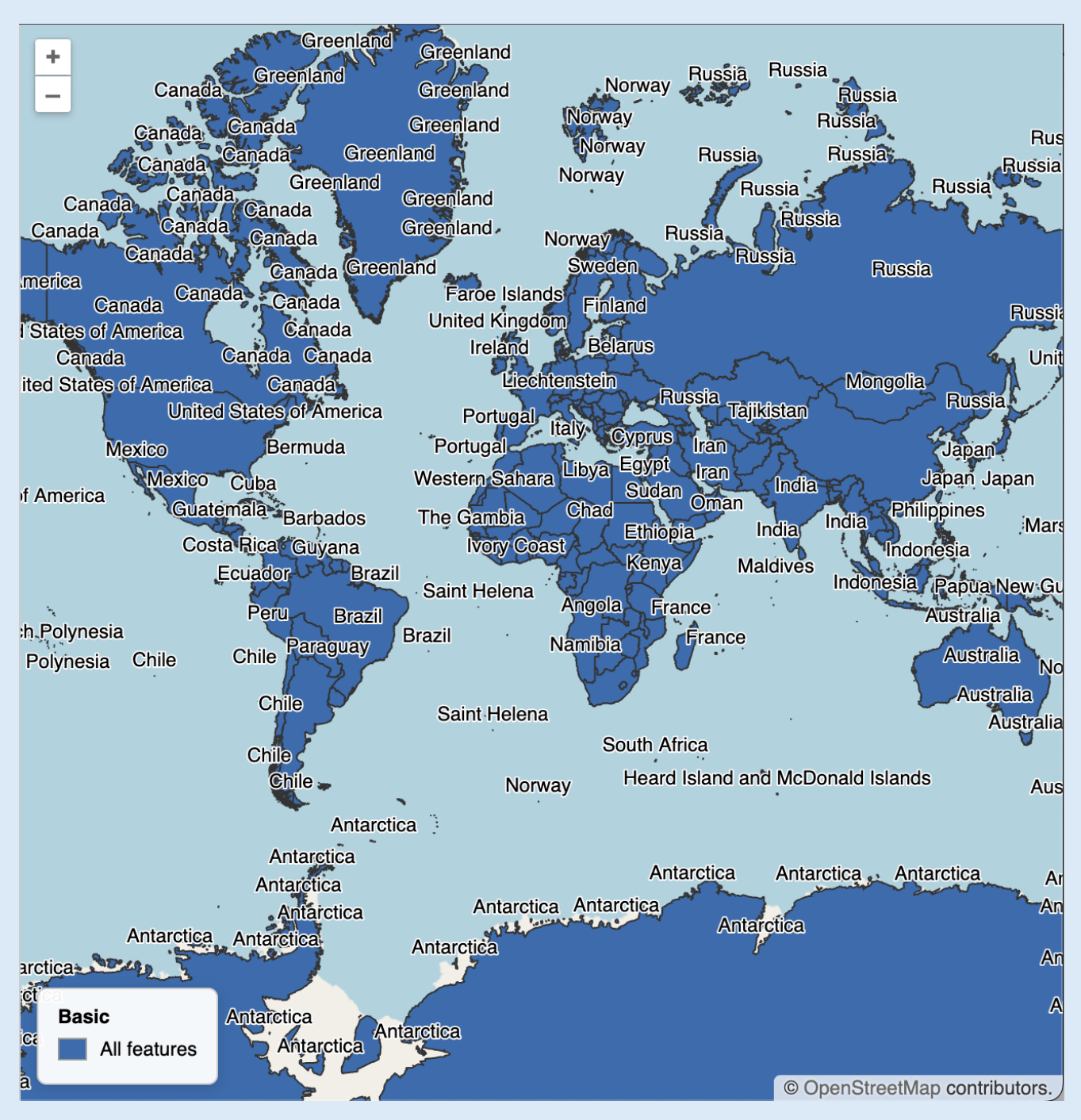
Graduated: Suitable color scheme for numerical values that are continuous, high number of unique values.

"Color countries with a graduated scheme based on population."



Label: Suitable for string attributes that consist of mostly unique values.

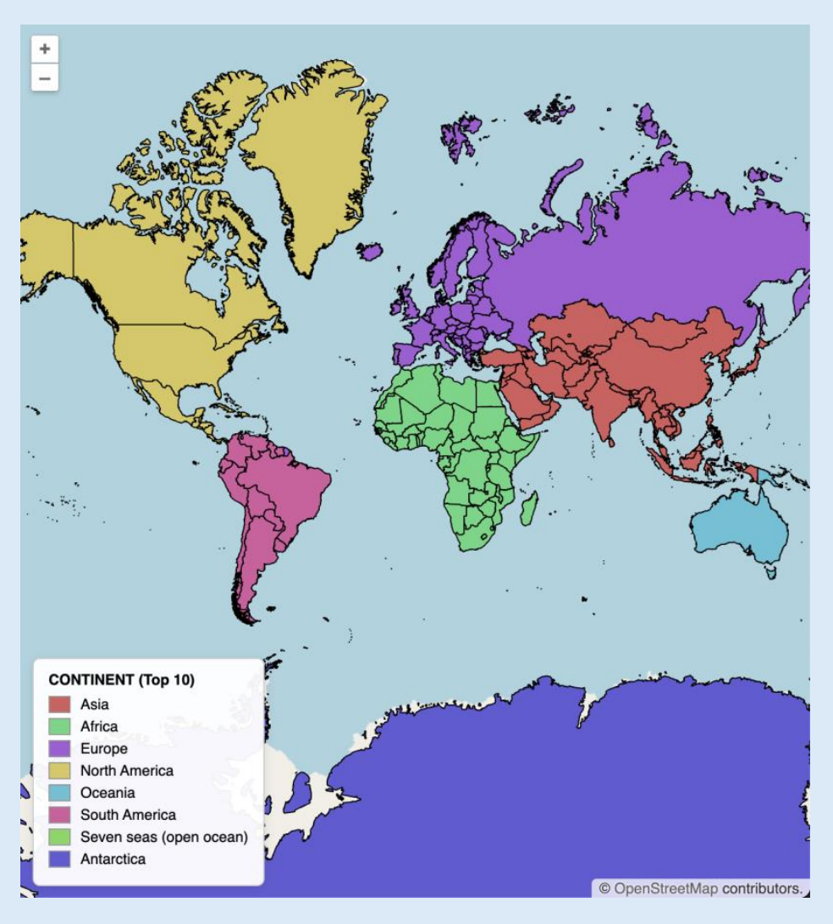
"Display country names as labels."



Basic: Default - If none of the other styles can be applied, e.g., all rows have same value.

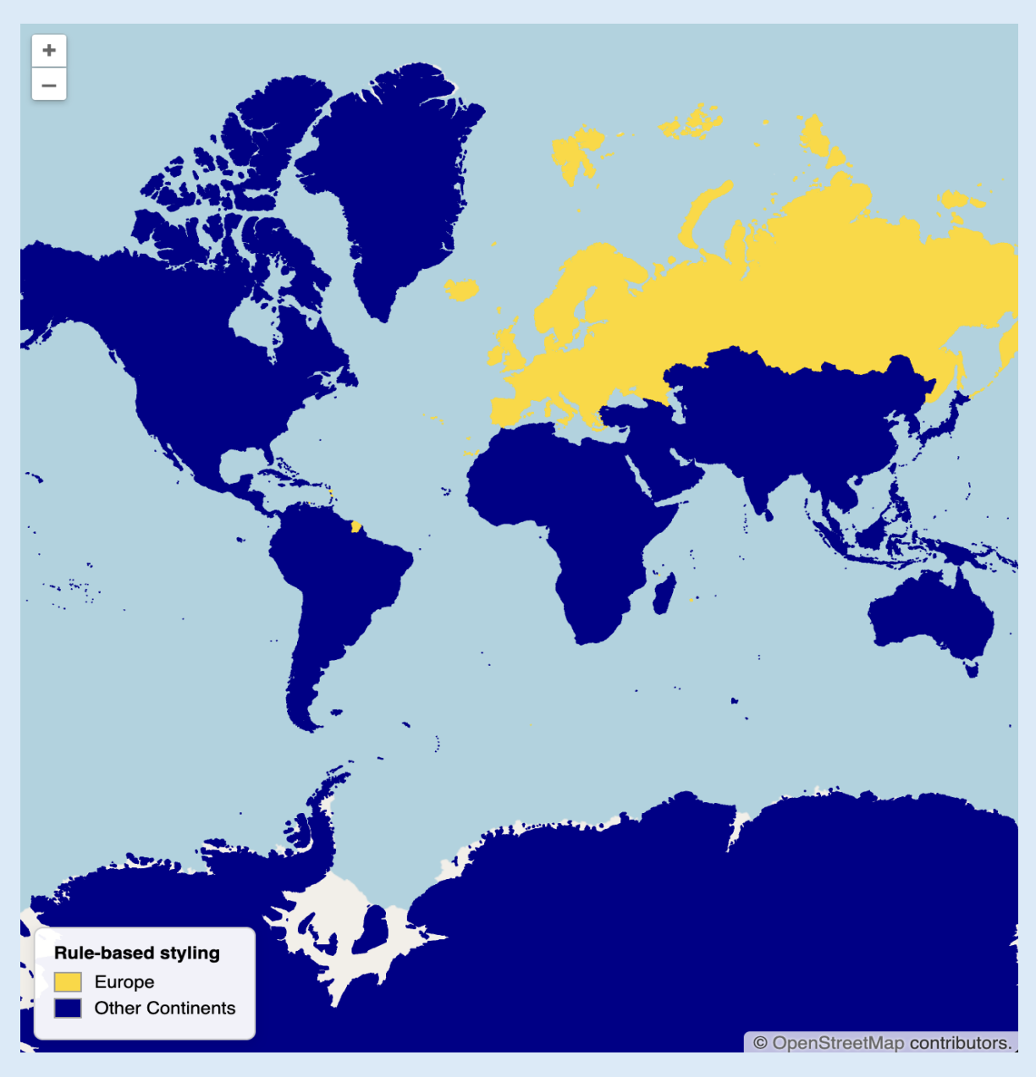
Categorized: Suitable for numerical / string values that have many repeated value.

"Color countries based on continent."



Rule-based: A certain value specified in user query. E.g., Show roads longer than 50 km in red.

"Show all countries in Europe in gold and all other countries with deep blue."



Example of rule-based LLM JSON response:

```
- when all:
attr: continent
op: ==
value: Europe
fill: #3B82F6
legend: Europe
- else:
 fill: #E5E7EB
 legend: Other
```

Conclusion and Future Work

Conclusion: LLMs can be used for exploring geospatial data. Because full datasets exceed per-call token limits and impose bandwidth, cost, or privacy overheads, we supply compact inputs: a small stratified sample, the parsed schema, and summary statistics (top-k, min/max, variance).

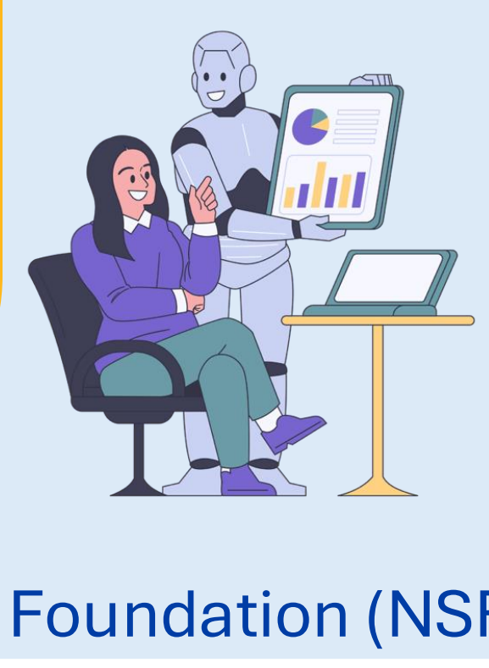
Interactivity is preserved via a low-latency two-prompt flow (attribute selection → style generation) enabling client-side rendering, and caches for instant restyling.

Scalability follows from exchanging aggregates (not raw tuples) and computing them server-side in SparkSQL.

Potential Future Work:

Multi-Attribute Styling: Combine attributes with layered encodings e.g., a categorical palette for one field plus a graduated ramp for another, and rule-based overlays.

Cross-Dataset Styling: Enable cross-layer styling via attribute/spatial joins and schema harmonization for combined analyses.



LEARN MORE
github.com/tarlaun/LASEK
Contact: tbaha001@ucr.edu

This work is supported in part by the National Science Foundation (NSF) under grant IIS-2046236