Lesson 8 – Spatial Data Integration

1 Overview

In this lab, we explore the power of **spatial data integration** — combining traditional geographic data (e.g., maps, shapefiles, satellite imagery) with data from **non-traditional sources**, such as:

- Social media feeds
- Tabular data (Excel, CSV)
- Mobility logs and sensor data
- Statistical datasets

Geolife GPS trajectory dataset

The **GeoLife** dataset, released by Microsoft Research, contains over **17,000 trajectories** collected from GPS-enabled devices in Beijing. Each trajectory consists of timestamped latitude and longitude points, often collected over months or even years.

- Captures individual human movement patterns.
- Contains metadata such as time, speed, and transportation mode.
- Ideal for trajectory analysis, stay-point detection, and space-time cube construction.

This data enables temporal-spatial modeling of urban mobility, urban behavior analysis or other related area.

Beijing Road Network

- Contains polyline geometries for different road types.
- Includes attributes such as road name, type (e.g., residential, primary), and directionality.

Weibo Check-in Data

Sina Weibo is one of the most popular Chinese microblogging platforms, often described as a hybrid of Twitter and Facebook. The dataset used here consists of **check-in records** collected from Sina Weibo during the year 2013. It includes a total of approximately **868 million check-ins** across **143,576 unique venues** throughout China. This type of data helps explore spatial social behavior, place popularity, and public space utilization from a human-centric perspective.

Since our source only includes point features, we **do not perform** natural language processing, geographic information extraction, and location encoding. Normally, the raw dataset of social

media is typically provided in `JSON` format, containing rich metadata including textual content, timestamps, user profiles, and full location information. An example of the original dataset can be found on Kaggle: `SocialNet Weibo Version 2`.

2 Setting Up the base Map & Georeference

- Create a new Map project in your geodatabase (New Project → Click 'Map').
- Name it `ex08`. Use the Projected coordinate system(e.g. WGS 1984 World Mercator[1]) for now(Right-click 'Map' → 'Properties' → 'Coordinate Systems' → Search 'Mercator' → Click CRS and then 'Apply').
- Add the connection to your data floder(**Insert** → 'Add Floder').

Now you can adjust the view to focus on the Beijing area.

3 GPS Trajectory data & Tabular Data

3.1 Importing the CSV and Project the feature

1. Use **Add Data** to select 'XYPoint Data', set *Input Table* as CSV path.
2. *Output Feature Class* → your name it. Keep default for *X Field, Y Field & Coordinate System*.
3. Click *Run* to import csv data as the Point format.
4. Open the *Project* tool, choose the layer you just add as the *Input Dataset or Feature Class*. Set the CRS of the 'Current Map (Map)' as the *Output Coordinate System*.
5. Click *Run* to project the csv data to the Projected coordinate system. Now you can remove the first layer you input to reduce memory pressure.

3.2 Describing the temporal data with elevation

The idea is the time dimension is encoded as elevation, allowing temporal patterns to be visualized in 3D space. By mapping time to vertical height, spatial-temporal data can be represented as a 3D structure — often referred to as a space-time cube(next part).

- Create a 3D view to the following work. **View** → **Convert** → 'To Local Scene' or 'To Global Scene'.
- Click **Map_3D** and Try to pan and zoom in 3D.
- Darg the projected point layer to the **3D Layers** to display.
- Right click the poiny layer, open the 'Properties' and click the 'Elevation'.

[1]The Mercator projection is mainly designed for global-scale applications. In everyday work, local projected coordinate systems are usually preferred to achieve higher spatial accuracy. For example, within Germany, the ETRS89 / UTM zone 32N (EPSG:25832) is commonly used, while in China, the CGCS2000 / Gauss-Krüger zones are typically applied.

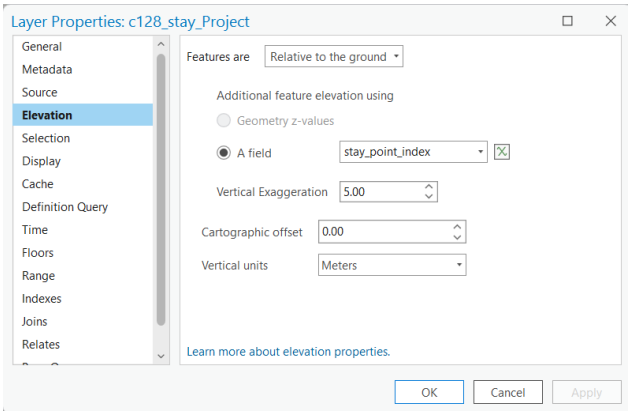


Figure 1: Elevation Menu

You can set the *A field* as some time-related field(like 'stay_point_index', or you can calculate your own filed based on time). Anyway, the result should be similar to Figure 2. You also can set symbology by right-clicking the point layer and click the *'Symbology'* for further process. In this figure, we set *Primary symbology* as 'Graduated Colors'[2]. Now you can try to find some patterns.

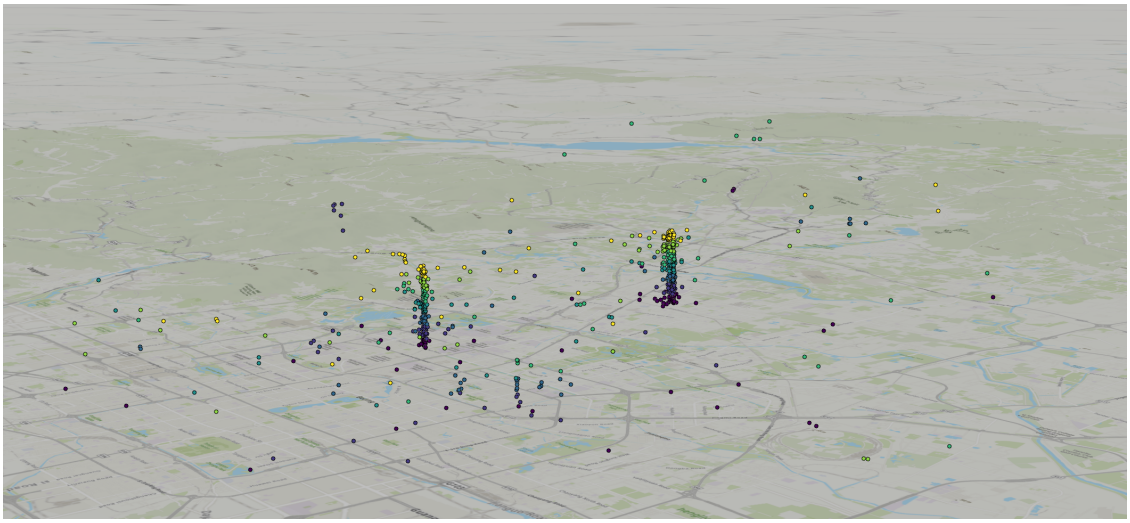


Figure 2: Demonstration of using elevation to represent space-time data.

3.3 Creating Space Time Cube

In this part, we aim to creat the standard space time cube for better description of the data.

1. Open the **Create Space Time Cube By Aggregating Points** tool.
2. *Input Features* should be the point layer, and name the ouput in *Output Space Time Cube*.
3. This tool allows us to aggregate points based on a specified time interval or spatial distance, helping to reduce memory usage and rendering pressure. Set the Time Field as 'arrival_time' or 'leaving_time'.Next, specify suitable values for the *Time Step Interval* and *Distance In-*

[2]You can improve the clarity of the visualization by using color to encode temporal information, such as assigning different years to different colors via field calculation.

terval[3].

4. Open the **Visualize Space Time Cube in 3D** tool, and set the *Input Space Time Cube* to the recently generated cube in your project folder. Set the *Cube Variable* to 'COUNT', and choose 'Value' as the *Display Theme*. Now you can see many white cubes in your screen.

5. It's difficult to observe meaningful patterns in the initial result, as shown in Figure 3. To improve clarity, right-click the cube layer and open the *Symbology* tool. Set the color of the first bin (representing zero values) to `No color`, as illustrated in Figure 4.

6. This adjustment results in a much clearer visualization, as shown in Figure 5.

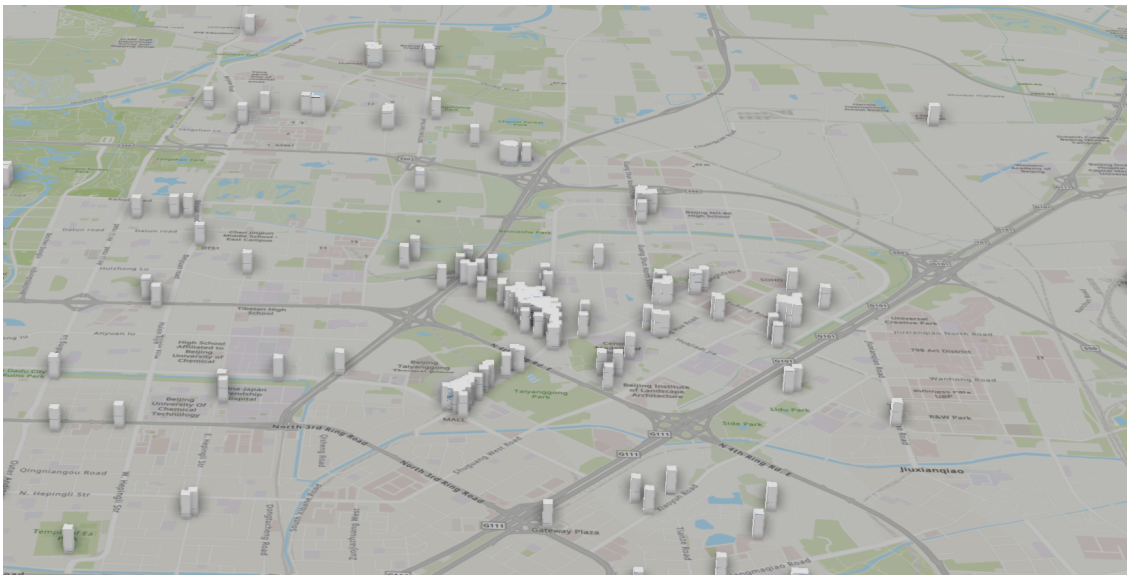


Figure 3: The initial visualization of Space time cube

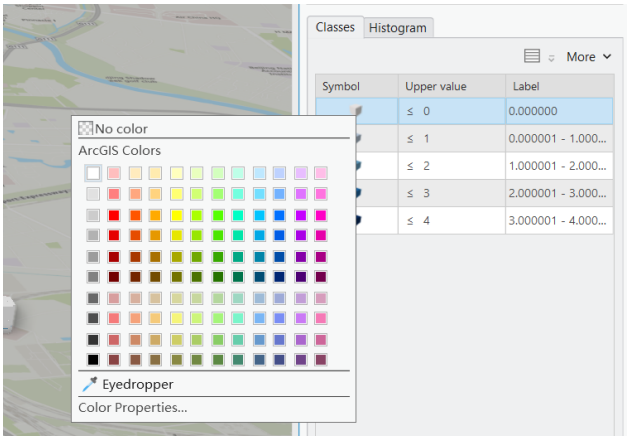


Figure 4: Change the color of the first bin

[3]Don't forget to select the appropriate unit & Avoid setting the interval too small, as this may cause the process to fail due to more than 2 Million Bins.

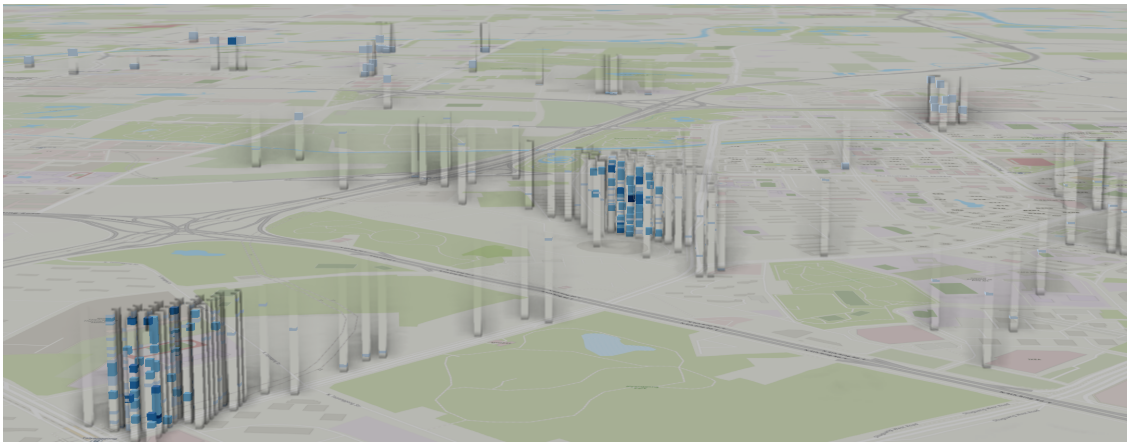


Figure 5: The improved visualization of Space time cube

4 Road Network & GPS trajectories

So far, our analysis has primarily focused on **stationary activities**, such as those occurring at home, in the office, or at school. However, spatial data integration also enables us to explore **mobility patterns** and **transportation behavior**. Now importing the road layer“ and the GPS point data as the last part showed.

4.1 Import & Project

- You can create a new project or just create a new map view for this part. Keep the GPS data as the point layer and import the provided the road layer.
- This layer primarily contains the main urban expressways in Beijing. Due to the city's unique spatial layout, these roads are commonly referred to as the *Second Ring Road*, *Third Ring Road*, and so on. By opening the **Attribute Table**, you can explore various associated attributes of each road segment as shown in Figure6.
- In this section, we aim to estimate a basic mobility indicator: the **average commuting speed**. As is commonly known, within a fixed distance, a larger number of GPS points generally indicates a lower relative speed, as it suggests the user moved more slowly through that segment. By examining the attribute table of the expressway layer, we can identify fields such as `length` or `shape_leng`, which likely represent the geometric length of each road segment. These values can serve as the denominator in our speed estimation calculation.
- You might be surprised to find that the values of the `length` or `shapeleng` fields are extremely small. This is because the road network dataset is currently stored in a **Geographic Coordinate System (GCS)**, specifically `WGS 84`. In GCS, the unit of measurement is not meters, but **decimal degrees**. This means the `Shape_Leng` field does not directly represent physical distances on the ground.[4]

[4]
- **Unit Inconsistency:** One degree of longitude or latitude corresponds to a different real-world distance depending on the geographic location.
- **Latitude Sensitivity:** At the equator, 1° of latitude is approximately 111 km. However, the longitudinal distance represented by 1° shrinks significantly as you move toward the poles.

Therefore, calculating meaningful speed or density metrics directly based on these values would be incorrect. Before performing any spatial analysis involving distances or areas, it is essential to **project the data into a projected coordinate system (e.g., UTM)** where the unit is in meters.

- Open the *Project* tool, choose the road layer you just add as the *Input Dataset or Feature Class*. Set the CRS of the 'Current Map (Map)' as the *Output Coordinate System*. Now you can see that a new field named `Shape_Length` has been added, which represents the length of each segment like Figure7

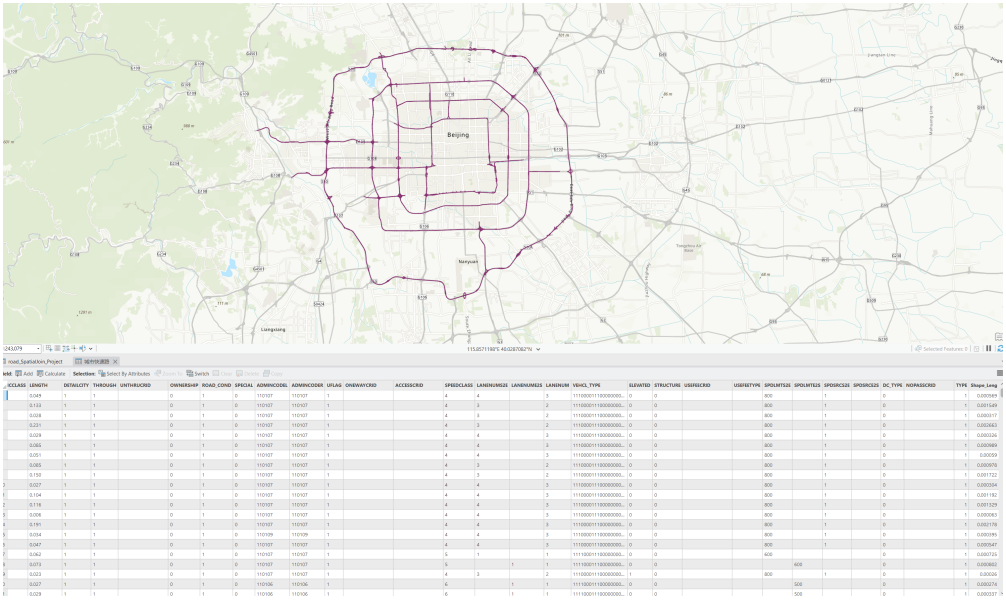


Figure 6: The preview of the road layer and its attribute table

TYPE	Shape_Leng	Longitude	Latitude	Altitude	DateTime	Density	Shape_Length
1	0.002177	116.357813	39.966376	0	2008/11/8 7:18:50	31.856535	242.336462
1	0.001481	116.309015	40.016525	0	2008/10/25 0:01:52	23.744687	165.931859
1	0.00004	116.368784	39.966668	0	2008/12/9 1:54:10	17.983917	4.448419
1	0.000051	116.368784	39.966668	0	2008/12/9 1:54:10	17.435995	5.735262
1	0.000054	116.368784	39.966668	0	2008/12/9 1:54:10	15.950617	6.26935
1	0.00028	116.310614	39.983988	0	2008/11/8 4:45:22	11.530957	31.220306
1	0.00028	116.343271	39.966202	0	2008/11/8 7:16:21	10.926031	31.118344
1	0.00043	116.392658	39.967229	0	2008/12/9 1:56:38	9.174792	47.957489
1	0.00014	116.368567	39.966669	0	2008/12/9 1:54:09	8.992158	15.569122
1	0.0003	116.355204	39.966285	0	2008/11/8 7:18:20	8.397705	33.342441
1	0.0002	116.405293	39.967626	0	2008/12/9 2:03:51	8.085636	22.261701
1	0.002258	116.326969	39.984692	0	2008/10/27 4:32:19	7.950907	251.543627
1	0.00016	116.327699	39.96625	0	2008/11/7 10:42:40	7.876971	17.77333
1	0.0003	116.34338	39.966415	0	2008/11/8 10:43:59	7.797869	33.342444
1	0.00025	116.339086	39.966251	0	2008/11/8 7:15:48	7.196306	27.792038
1	0.00063	116.397374	39.967399	0	2008/12/9 2:01:17	7.127969	70.146211
1	0.000439	116.355597	39.966269	0	2008/11/8 7:18:24	6.951307	48.911663
1	0.00004	116.347312	39.966249	0	2008/11/8 7:17:14	6.883971	5.8106
1	0.000669	116.34609	39.966265	0	2008/11/8 7:17:02	6.713257	74.479493
1	0.00071	116.391974	39.967216	0	2008/12/9 1:56:16	6.575514	79.081268
1	0.00033	116.35566	39.96651	0	2008/11/8 10:41:57	6.541515	36.688747

Figure 7: `Shape_Length` refers to the length of the line segment.

4.2 Spatial Join

As the theme *Spatial Data Integration* suggests, the key challenge when working with heterogeneous data sources lies in effectively merging them. In this case, we aim to integrate GPS trajectory points into the road network by transforming them into meaningful road-related attributes.

1. Open the *Geoprocessing* pane and search for the `Spatial Join` tool.
2. Set the parameters as follows:
   - *Target Features*: Select your road network layer.
   - *Join Features*: Select your GPS points layer.
   - *Output Feature Class*: Specify a name for the output, e.g., `Roads_with_GPS_Count`.
   - *Join Operation*: Choose `JOIN_ONE_TO_ONE`.
   - *Match Option*: Select `WITHIN_A_DISTANCE`.
3. *Set Search Radius:* Choose a reasonable distance such as `20 meters`. Since GPS points rarely fall exactly on the centerline of roads, this buffer helps "capture" nearby points. Adjust the distance based on the accuracy of your GPS data.
4. *Run the tool.* After completion, a new feature class (e.g., `Roads_with_GPS_Count`) will be created.
5. *Inspect the result:* Open the attribute table of the new layer. As Figure8 shows, you will find a new field (usually named `Join_Count`) indicating how many GPS points are within the specified distance of each road segment.

	OBJECTID_1 *	Shape *	Join_Count	TARGET_FID	OBJECTID
1	1273	Polyline	386	1272	1505
2	5858	Polyline	197	5857	6090
3	3631	Polyline	4	3630	3863
4	3630	Polyline	5	3629	3862
5	3632	Polyline	5	3631	3864
6	1193	Polyline	18	1192	1425
7	4416	Polyline	17	4415	4648
8	2835	Polyline	22	2834	3067
9	4477	Polyline	7	4476	4709
10	4394	Polyline	14	4393	4626
11	3654	Polyline	9	3653	3886
12	4224	Polyline	100	4223	4456
13	5821	Polyline	7	5820	6053
14	4414	Polyline	13	4413	4646
15	5823	Polyline	10	5822	6055
16	2810	Polyline	25	2809	3042
17	855	Polyline	17	854	1087
18	5820	Polyline	2	5819	6052
19	4239	Polyline	25	4238	4471
20	4556	Polyline	26	4555	4788
21	4393	Polyline	12	4392	4625

Figure 8: `Joint Count`

4.3 Normalize Point Density by Segment Length

A common pitfall in raw point counting is that longer segments naturally collect more GPS points. For example, a 1 km road segment with 50 points and a 100-meter segment with 30 points would have `Join_Count` values of 50 and 30, respectively — but clearly, the shorter segment has a higher

intensity. To address this, we need to calculate a **density metric** by normalizing point count with segment length.

1. *Add a new field:* In the attribute table of `Roads_with_GPS_Count`, create a new field named `Density`. Set the data type to `Double` (floating-point).
2. *Open the Calculate Field tool:* Right-click the new `Density` field and choose `Calculate Field`[5].
3. *Enter the calculation expression:* For basic density (points per unit length):

   ```
   !Join_Count! / !Shape_Length!
   ```

4. If you prefer to express it as points per 100 meters[6]:

   ```
   (!Join_Count! / !Shape_Length!) * 100
   ```

4.4 Visualization

1. Right-click on the `Roads_with_GPS_Count` layer and select **Symbology**.
2. In the *Primary symbology* panel, choose **Graduated Colors**.
3. For the *Field*, select the previously calculated `Density` field.
4. Choose a *Color scheme* that resembles a heatmap to highlight intensity[7].
5. The visualization result is like that shown in Figure 9.

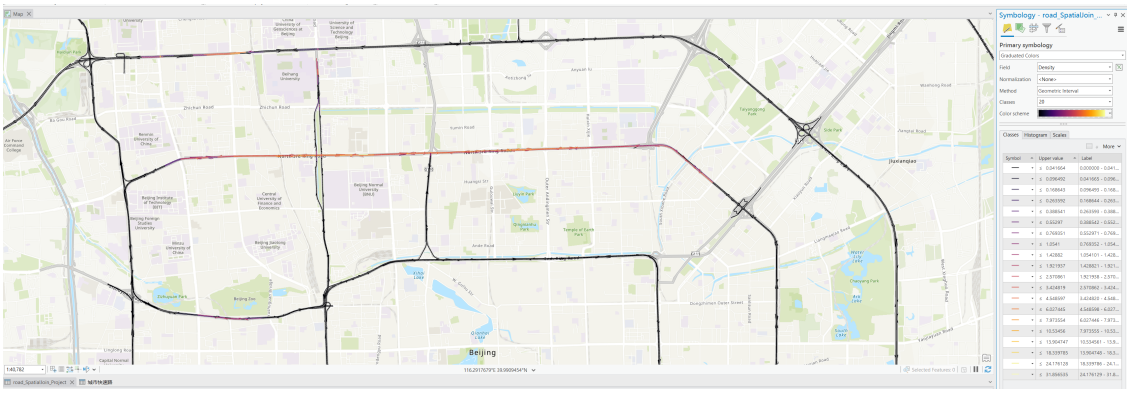


Figure 9: `Heap Map according to the Density`

[5]Note: Make sure to handle cases where `Density = 0` to avoid division by zero.

[6]In an ideal situation, the **inverse of density** can approximate relative speed. Add a new field, e.g., `Speed_Approx`, and use: ‘1 / !Density¡

[7]To emphasize high-density segments, consider adjusting line width using **Graduated Symbols** in combination with color, making hotspots appear thicker and more prominent.

5 Social Media Data

5.1 Preview

This layer contains approximately 140,000 records, each with corresponding coordinates and a wide range of attribute information. As shown in Figure10, the data appears to be well integrated with the basemap. Aside from the coordinate system still being in the geographic coordinate system `WGS 84` (which needs to be projected), it seems ready for use.

The original author mentioned that georeferencing is required but did not specify the extent of the positional deviation. However, if you zoom in on the layer, you will notice that many points fall into rivers or lakes, which clearly indicates spatial inaccuracies. This suggests that the dataset has significant coordinate errors and cannot be used directly for precise spatial analysis.

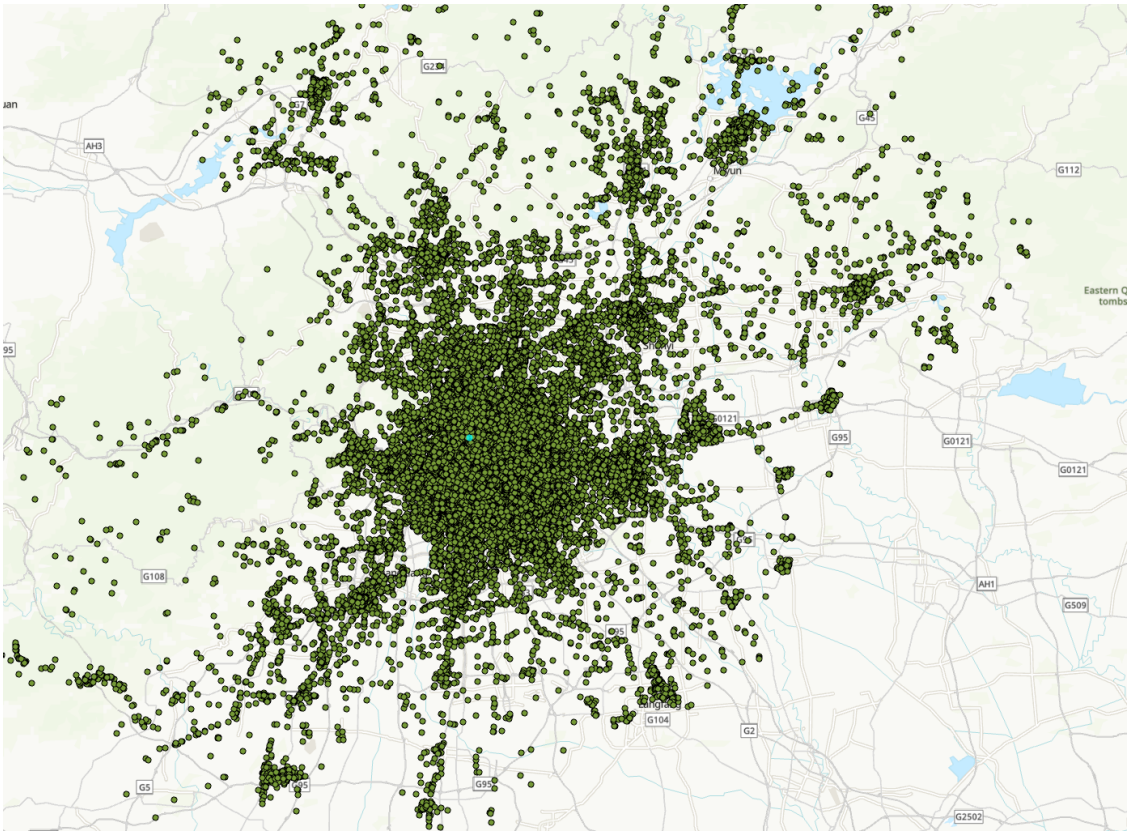


Figure 10: Weibo check-in data

5.2 Georeferencing

Georeferencing refers to the process of associating images, maps, or other data with geographic coordinates in the real world, and is commonly used in fields such as Geographic Information Systems (GIS) and 3D modeling.

In this step, we aim to **adjust the spatial accuracy of the point data**. As noted by the original data providers: The coordinates of this data have been modified officially. Additional georeferencing might be needed. Although the dataset appears to be in the `WGS 84`, there exists a notable positional bias in the actual point locations. This discrepancy is likely due to intentional

coordinate obfuscation for privacy, national security, or commercial reasons — a common practice in publicly released spatial data, especially from commercial map services.

Your task is to perform **georeferencing** by matching a set of sample points with their *known coordinates*[8], and then apply an appropriate spatial transformation to align the entire dataset accordingly.

5.2.1 Importing & Linking target coordinates

To perform spatial adjustment between two sets of control points (source and target), follow the steps below to create spatial adjustment links based on matching IDs.

- Import the CSV data as the XY format(*Add Data-> XY Point Data*). This is target point layer containing 5 known points.
- In the *Geoprocessing* pane, search for and open the *Join Field* tool.
- *Input Table*: Select the layer you just add. *Input Field*: Select the `FID`. *Join Table*: Select the `weibo_points` layer. *Join Field*: Also select the `FID`.
- *Transfer Fields*: Select the `X` & `Y`. Other options keep default and run.
- Open the *Attribute Table* of the control points layer, there are two new fields from the weibo point layer like Figure11
- Open the *XY to Line*, select this joined layer as Input Table. Set the pareamters as shown in Figure12[9].

	OBJECTID *	Shape *	FID	x_target	y_target	x_src	y_src
1	1	Point	274	39.890662	116.638994	39.891899	116.643997
2	2	Point	289	39.952412	116.329021	39.958302	116.339996
3	3	Point	22288	39.902648	116.631173	39.903999	116.637001
4	4	Point	34133	39.947533	116.421081	39.948799	116.427002
5	5	Point	34345	39.956339	116.347375	39.9575	116.352997
	Click to add new row.						

Figure 11: Attribute Table after `Join Field`

[8]Unlike satellite imagery, where georeferencing is relatively straightforward due to clearly identifiable landmarks, georeferencing textual data (such as location names extracted from social media) presents more challenges. In satellite images, it's often easy to find corresponding reference points on a map; however, for place names or venues mentioned in text, identifying accurate control points can be difficult.

In this case, our approach is to locate venues that still exist today—preferably small or well-defined places to reduce potential misalignment. And then manually find their corresponding coordinates using the *basemap* in ArcGIS Pro. These coordinates are then recorded as ground control points for georeferencing.

While this method is time-consuming and yields only a limited number of control points with moderate accuracy, the purpose of this experiment is to understand the concept of georeferencing rather than to pursue high precision.

[9]In GIS tools, `X Field` refers to longitude and `Y Field` to latitude. In this case, `X Field` = `y` and `Y Field` = `x`. This swap is easy to miss.

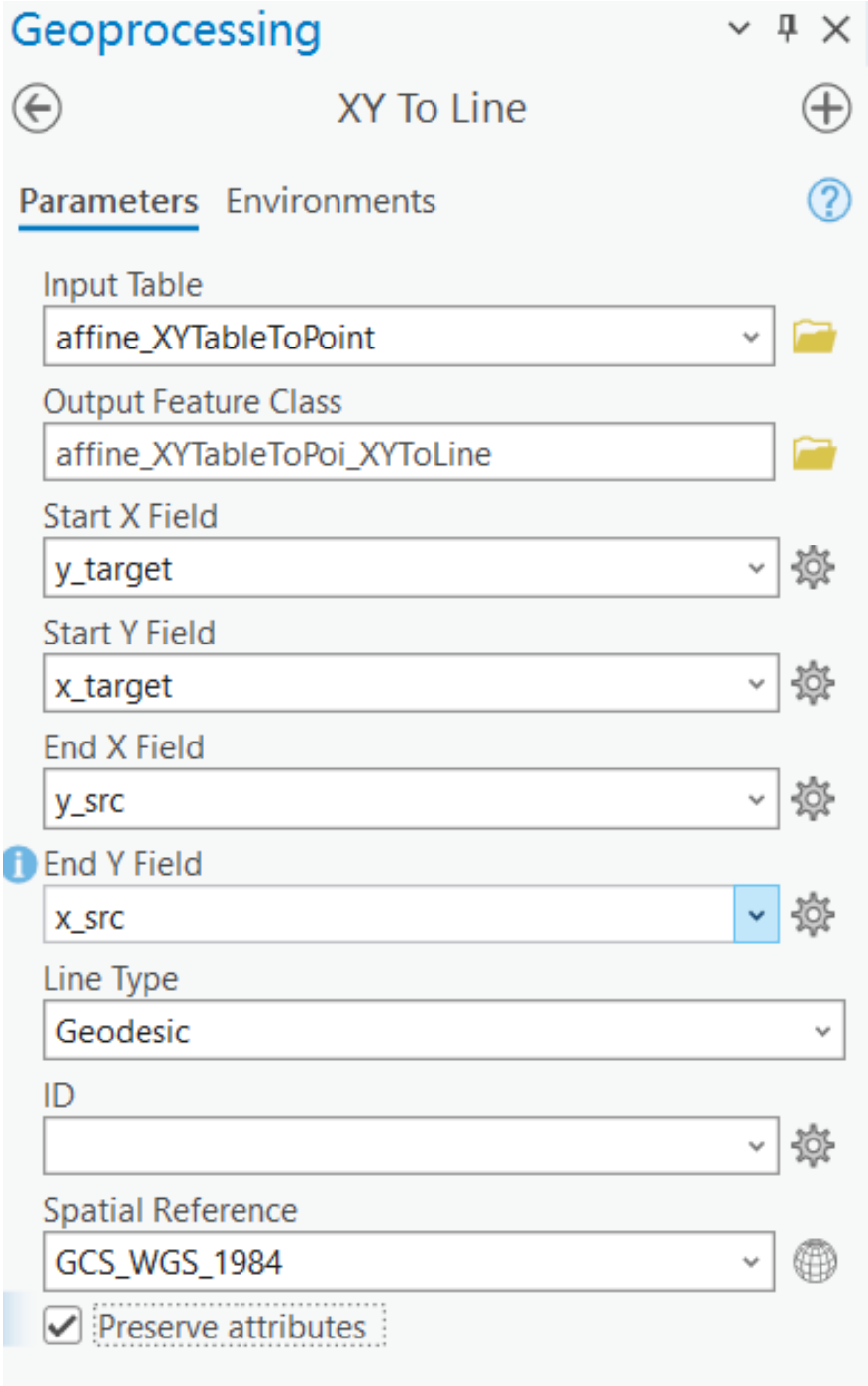


Figure 12: XY To Line

5.2.2 Transforming Features

- In the *Geoprocessing* pane, search for and open the *Transform Features* tool(Figure13).
- *Input Features*: Select the `Weibo_points` layer[10], whose coordinates will be transformed.

[10]This operation will modify the original layer. Make sure to back up your data or save a copy of the project before proceeding.

- *Input Link Features*: Select the `Line` Feature Layer as the above processing.
- *Method*: `Affine transformation`.
- *Output Link Table*:You name it[11].

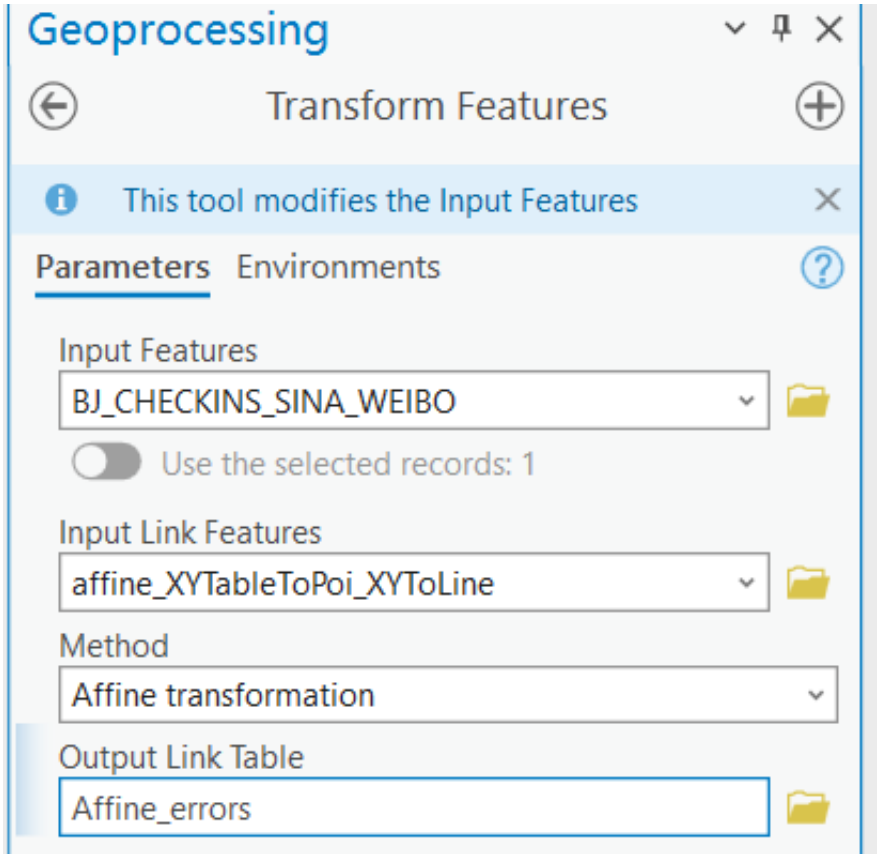


Figure 13: Transform Features using Affine Transformation

Once the tool has been executed, the coordinates will be georeferenced and ready for further analysis, You can either save it as a new point feature or continue working with it in this project. As illustrated in Figure14, the transformation does not result in zero error due to the use of redundant control points. With a greater number of accurate control points, the coordinate accuracy can be assessed using the *Root Mean Square Error (RMS)* metric or other algorithms.

	OBJECTID *	Orig_FID	X_Source	Y_Source	X_Destination	Y_Destination	Residual_Error
1	1	1	116.643997	39.891899	116.638994	39.890662	0.000968
2	2	2	116.339996	39.958302	116.329021	39.952412	0.003568
3	3	3	116.637001	39.903999	116.631173	39.902648	0.001381
4	4	4	116.427002	39.948799	116.421081	39.947533	0.00087
5	5	5	116.352997	39.9575	116.347375	39.956339	0.003113
	Click to add new row.						

Figure 14: Transform Features using Affine Transformation

[11]The output table containing the input links and their residual errors.

5.3 Integration with Road Layer

Next, we can apply the same integration process to the georeferenced social media data like what you did in Section4, including performing spatial join and visualization. In this task, since the social media data lacks temporal information, we will not focus on time-dependent topics such as traffic analysis. Instead, we focous on spatial distribution. We make a simple assumption that the check-in data contains a large number of tourist records. Based on this, we can use the data in combination with the road network[12] to identify hotspots of tourist attractions.

As shown in Figure15, and based on the basemap, we can observe several clear patterns. First, the areas near the ring roads and major intersections appear as high-density hotspots. In particular, the eastern central part — where the Beijing Railway Station and the CBD (Central Business District) are located — also exhibits strong intensity. Additionally, the upper-left (northwestern) area shows generally higher density, which corresponds to Beijing's "Silicon Valley", home to more than a dozen top universities and leading tech companies.

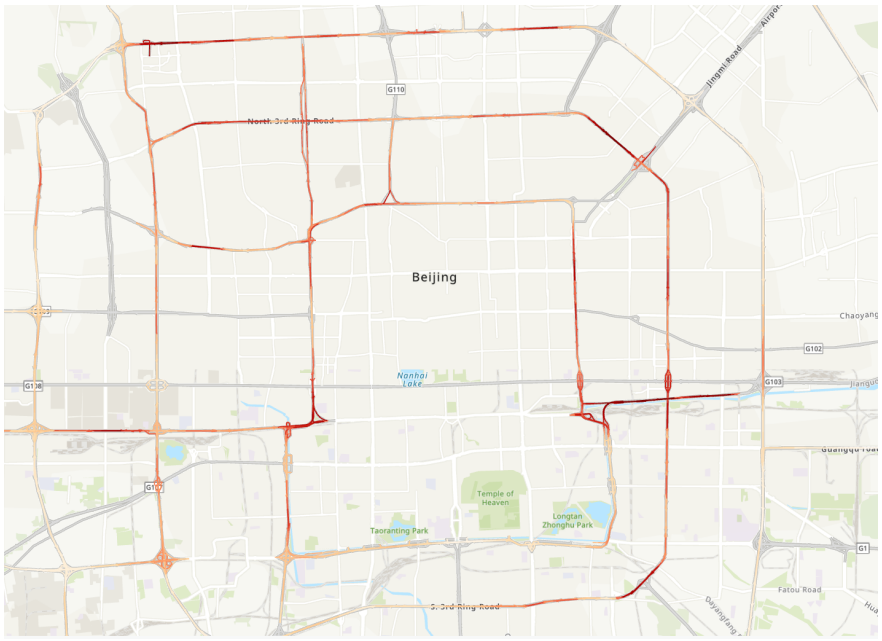


Figure 15: Road heat map according to the soical media data

Combining this heatmap with spatial patterns, it appears that tourists are not the dominant group represented in the Weibo check-in data. Instead, a significant portion of the check-ins likely comes from students, office workers, and local residents[13].

6 Furthermore

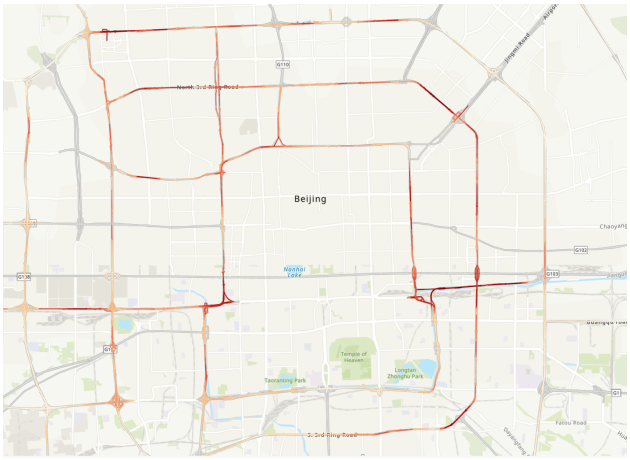
A simple comparison can also be made between GPS and Weibo check-in data. By applying *kernel density estimation (KDE)*, spatial hotspots for each dataset can be derived and compared. As this

[12]Beijing's expressways are the city's most important ring roads, and many landmarks are typically located along them, for example, Bird's Nest & Water Cube, Yiheyuan, Temple of Heaven. Here you can set a larger distance threshold like 0.7 - 1.5km during the spatial join

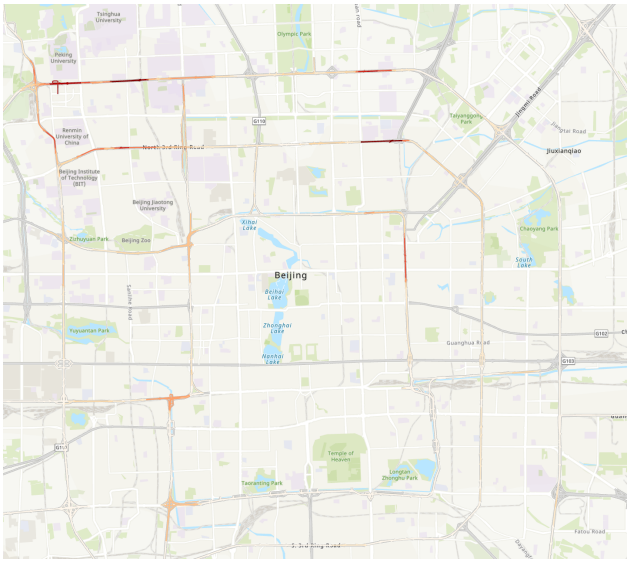
[13]This is not a definitive conclusion, but it conveys an important idea: In spatial analysis, you can begin with a hypothesis based on intuition or perception. Then, we apply scientific methods to either support or challenge that hypothesis with empirical evidence.

is more about the field of spatial analysis, further discussion is beyond the scope of this task.

However, through integration with the road network, we can observe that the GPS data is highly concentrated around university areas in Beijing. This is quite understandable: if you are familiar with the concept of *survivorship bias*[14], this serves as a classic example. The dataset was collected by university professors and researchers using wearable GPS devices (see the dataset description for details); therefore, most volunteers were university staff and students. As a result, data from other professions and demographics are relatively underrepresented.



(a) Weibo Check-in Data



(b) GPS Trajectories Data(84 Users)

Figure 16: Comparison of Heat Maps

[14]Survivorship bias or survival bias is the logical error of concentrating on entities that passed a selection process while overlooking those that did not. See video or wiki for more details

Contents