Interview Question: GPS Data & Map Building Prioritization

You're given **gnss_data_jabotinsky_7_8.jsonl** - a 31-minute GPS trace with 2,219 readings from a camera mounted on a vehicle.

Your company builds Google Street View-style mapping. Processing all captured footage is expensive, so we need to **prioritize which time segments are most valuable for building the map**.

Part 1: Visualization (15 minutes)

Task: Create a visualization to explore the data.

Plot the GPS trajectory in a way that helps you understand the vehicle's behavior over time.

Deliverable: One or more plots showing the trajectory.

Part 2: Maximum Value Window (30 minutes)

Task: Write a function that finds the **most valuable S-second window** for map building.

```
def extract_max_value_window(input_file, duration_seconds, output_file):
"""
Find the S-second window most valuable for building the map.
Returns metadata dict with your value metric and justification.
"""
```

Requirements:

- Implement a sliding window to scan the entire dataset
- Define "map value" - choose a metric that makes sense for street mapping
- Output a JSONL file with just that time window
- Return metadata explaining your choice

Test your function with:

- duration_seconds=120 (2 minutes)
- duration_seconds=300 (5 minutes)

Most Important: Write 2-3 sentences explaining why your metric correctly identifies the most valuable segments for building a street map.

Part 3: Pattern Recognition (15 minutes - BONUS)

Task: Analyze the window from Part 2. What pattern do you see?

Choose either:

- **Visual:** Plot it and describe the pattern in words
- **Programmatic:** Write code to classify the movement type

Example patterns: straight line, zigzag, loop, stop-and-go, random walk

Output: A description or classification of what the vehicle was doing.

What We're Evaluating

Part 1:

- Can you quickly explore unfamiliar data?
- Do your visualizations help you understand the problem?

Part 2:

- **Critical:** How do you define "map value"?
- **Critical:** Why is your definition useful for street mapping?
- Does your code work correctly?
- Do you handle GPS noise (~0.7m accuracy)?

Part 3:

- Can you extract meaningful insights from the data?
- Do you connect patterns back to the practical use case?

Tips

■■ **Time management:** Part 2 is the core - don't spend too long on Part 1

■ **Use the data:** Run your code, don't just theorize

■ **Justify choices:** "Why" matters as much as "what"

■ **Pragmatic over perfect:** Working code with good reasoning beats over-engineered solutions

---

You have 60 minutes. Good luck!