Use of GenAI

I provided this prompt to GenAI: "I have a large collection of GPS files in NMEA format stored in text files. Each GPS file contains longitude, latitude, speed, timestamp, and altitude measurements. These GPS files are spread across 60-90 different text files with a sampling rate of every half second. I don't need precision shorter than every 5 seconds. I want to create a KML file (Keyhole Markup Language) that I can use in Google Earth to visualize the data. I want to see a white line representing the convex hull of all trips, and yellow markers where there are altitude peaks. The data includes stops for gas, groceries, and errands, but I don't care about these stops - I just want the convex hull. Can you help me write a Python 3 program to parse all these files?"
After receiving the initial code that used pynmea, I encountered datetime errors and asked GenAI to explain what these errors meant.
I also asked about real-world methods for checking data authenticity, and GenAI explained checksum calculation techniques.

What I needed to do to get my parser working

What was needed:

- Used the pynmea library with basic understanding of NMEA format structure (including usage of 'A', '1', '*' markers)
- Implemented regex pattern matching for data validation
- Applied checksum calculation for data integrity verification

Challenges encountered (based on file 2025_04_10__220057_gps_file.txt):

1. Data format misunderstanding: The biggest challenge was incorrectly assuming the file structure. Initially, I tried parsing all lines as raw NMEA sentences using pynmea2, which only extracted 1 GPS point from 10,780 lines of data.

- **What went wrong**: The file actually contained mixed formatting where every third line had pre-parsed GPS data (starting with "lng="), making standard NMEA parsing ineffective for most of the data.

2. DateTime operation error: Encountered this error: "unsupported operand type(s) for -: 'datetime.time' and 'datetime.time'"• **What went wrong**: The _should_sample method tried to subtract datetime.time objects,
but Python's subtraction operator only works with datetime.datetime objects.

3. KML visualization issue: The generated KML file only showed a blank Earth view in Google Earth.

- **What went wrong**: KML format expects coordinates in lon, lat, altitude order, but I was providing them in lat, lon, altitude order.

Conclusion

GenAI cannot be relied upon alone for complete code solutions, as there will always be issues requiring critical thinking. For example, the given dataset contained both raw NMEA sentences and pre-parsed data lines.
Data analysis should never begin without first examining how the actual data looks and understanding its structure.
GenAI is extremely valuable for learning purposes and explaining concepts.
Libraries like pynmea cannot be solely relied upon since real-world data often has inconsistencies. Edge cases must be designed using domain knowledge of the specific dataset. While this dataset used standard GPS talker ID "GP" followed by "RMC/GGA," other datasets might use different formats that pynmea might not recognize.
The should_sample() error highlighted the importance of careful implementation when performing arithmetic operations on different data types.
The KML parameter issue demonstrated the need for careful examination of program input requirements and parameter ordering.