ChatGPT Usage Transparency Report

1. Usage Overview

This project 'Gemscap' was developed leveraging the capabilities of Large Language Models (LLMs) to accelerate the software development lifecycle. The AI acted as a technical consultant and pair programmer, assisting in architectural decisions, code optimization, and rapid debugging, while the core logic and system design were directed by me.

2. Understanding Financial Terms

The AI assisted in clarifying and implementing complex quantitative finance concepts:
- **Hedge Ratio (OLS & Huber)**: Calculated to neutralize the market risk of one asset against another. While OLS minimizes squared errors, Huber Regression was suggested by the AI to provide robustness against market outliers and sudden volatility spikes.
- **Cointegration vs. Correlation**: clarified that high correlation implies assets move together, but cointegration implies they share a long-term equilibrium. This distinction is the foundation of the pairs trading strategy implemented here.
- **Z-Score & Mean Reversion**: Used to normalize the spread. A signal is triggered when the z-score exceeds a threshold (e.g., +/- 2.0), indicating the spread is statistically stretched and likely to revert.
- **Half-Life**: The time it takes for the spread to revert halfway to its mean. This metric, derived from the Ornstein-Uhlenbeck process, was implemented to determine the optimal holding period and check if the mean-reversion speed is profitable.
- **Stationarity (ADF Test)**: The Augmented Dickey-Fuller test ensures the spread has constant statistical properties over time, a prerequisite for valid statistical arbitrage.

3. Tech Stack Selection

AI Recommendations for a high-frequency capable architecture:
- **Backend**: FastAPI (Python) - Recommended for its native `asyncio` support to handle concurrent WebSocket streams and background analytics tasks without blocking.
- **Frontend**: React (Vite) - Chosen for its Virtual DOM efficiency in rendering high-frequency data updates and the rich ecosystem of charting libraries like Recharts.
- **Database**: SQLite (WAL Mode) - Selected as a lightweight, serverless option. The 'Write-Ahead Logging' mode was specifically suggested to handle concurrent reads/writes during real-time data ingestion.

ChatGPT Usage Transparency Report

4. Error Resolution with AI

Significant technical hurdles resolved through AI consultation:
a) **500 Internal Server Error (Race Condition)**: Diagnosed a race condition where the server reloaded during a frontend polling request. Implemented robust exception handling and file-based logging to capture tracebacks.
b) **NumPy Serialization**: Identified that native NumPy data types (int64, float32) crash the JSON serializer. A custom `recursive_sanitize` function was co-developed to safely convert these to Python primitives.
c) **CORS Policy Blocking**: Resolved cross-origin resource sharing issues between the React frontend (port 5173) and FastAPI backend (port 8000) by correctly configuring `CORSMiddleware`.
d) **React Hook Dependencies**: Fixed `useEffect` infinite loops by correctly identifying missing dependency arrays, ensuring the dashboard only refreshes when configuration state changes.
e) **Pandas SettingWithCopyWarning**: Clarified the difference between creating a view vs. a copy of a DataFrame, leading to cleaner and warning-free data manipulation code.

5. Conclusion

The integration of Generative AI into the development workflow significantly reduced the time-to-market for Gemscap. Rather than writing every line of boilerplate, I could focus on high-level strategy and system integrity. The result is a sophisticated, production-grade quantitative dashboard built with speed and precision, demonstrating the power of human-AI collaboration in modern software engineering.