expected_returns analysis

The `expected_returns` module in the PyPortfolioOpt library is designed to estimate the expected returns of assets, which is a fundamental input for portfolio optimisation and risk management.

Let's dive deeper into the purpose, functionality, and the pros and cons of the methods used, and of the module as a whole.

• • •

Purpose of the `expected_returns` Module

1. **Estimating Future Returns**: The primary purpose is to provide various methods to estimate the future returns of assets based on historical price data.
2. **Versatility in Return Estimation**: The module offers different methods to calculate expected returns, signalling that no single method is universally best. Varying financial scenarios and market conditions might favour one method over another.

Key Functions and Their Methodologies

Elaborated deeply through the analysis document on github.

General Functionality of the `expected_returns` Module

Pros:

- Functions are designed to work with pandas DataFrames, the industry-standard in handling financial data in Python.
- Offers options like log returns, compounding, and frequency adjustments to tailor the calculations to specific needs/requirements.
- Includes checks for NaNs and infinite values to ensure data integrity, and avoid errors.

Cons:

- Most methods rely heavily on historical data, which may not always be a reliable indicator of future performance.
- Methods like CAPM are based on theoretical assumptions that might not hold in all market conditions.

Conclusion

The `expected_returns` module in PyPortfolioOpt is a powerful tool for estimating the future returns of assets, offering a range of methods to suit different investment strategies and market conditions. While it provides flexibility and integration with the Python data analysis ecosystem, we must be mindful of the limitations and assumptions inherent in each method.
