Aedan W. Chiari

aedan.chiari@gmail.com | 802-777-8390 | Denver, CO | linkedin.com/in/aedan-chiari
Python • Rust • SQL • Dagster • Polars • DuckDB • Pandas • Delta Lake • Kubernetes • Bloomberg

EDUCATION & CERTIFICATIONS

University of Southern California, Marshall School of Business | May 2023
BS Finance (3.75 GPA, Magna Cum Laude) | Minor: Applied Data Analytics
FINRA Series 65

PROFESSIONAL EXPERIENCE

SitusAMC | Denver, CO

Quantitative Developer | Hedge Advisory | Aug 2023 - Present

- Built production analytics platform in Python processing 1M+ loan records daily with 99.9% uptime, delivering automated dashboards and risk reports within 30-minute SLAs for pre-market hedge decisions
- Achieved 75% faster computation through Python infrastructure modernization: migrated from Access/Excel to Polars/DuckDB with Arrow zero-copy processing, implemented Delta Lake on S3 for versioned storage with ACID guarantees, and deployed distributed task execution using Celery/RabbitMQ
- Built real-time market data pipeline with Dagster ingesting Bloomberg/CME/Refinitiv feeds to generate yield curves, volatility surfaces, and portfolio risk metrics
- Deployed containerized applications to Kubernetes using Docker and Azure DevOps CI/CD pipelines
- Built systematic hedging strategies using interest rate swaps, treasury futures, TBAs, and options to manage duration and convexity risk, rebalancing positions based on cash flow dynamics and volatility
- Developed PnL attribution framework enabling systematic analysis by decomposing returns into market movement, hedge effectiveness, and carry across $50m+ multi-instrument books
- Mentored junior analysts on Python data engineering best practices (Dagster orchestration, DuckDB optimization), enabling delivery of production pipelines reducing manual reporting time by over 60%

PERSONAL PROJECTS

Derivatives Pricing Library | github.com/aedan-chiari/rust-quant

- Built high-performance derivatives pricing library in Rust with Black-Scholes option pricing, full Greeks calculation (delta, gamma, vega, theta, rho), and yield curve bootstrapping with multiple interpolation methods
- Achieved 10-30x speedup over Python implementations through AVX2 SIMD vectorization and Rayon parallelism, with zero-copy PyO3 bindings for seamless integration
- Designed type-safe functional API with comprehensive test coverage validating against market-standard models