The Journal of Open Source Software

Bruno: A Julia package for simulation, financial asset pricing and delta hedging

Mitchell Pound [1], **Spencer Clemens** [1], **Tyler Brough** [1], **Pedram Jahangiry** [1], **and Janette Goodridge** [1]

1 Utah State University

DOI: 10.21105/joss.05056

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @lrnv
- @omendezmorales

Submitted: 14 December 2022
Published: 20 February 2023

Summary

When engaging in activities in financial markets, market makers and other financial practitioners face a variety of risks. Many attempt to reduce these risks by hedging. Hedging is entering an offset position in an investment or asset, allowing potential risks to be mitigated and transferred to investors willing to take the risk (Culp, 2011). Hedging is often accomplished using financial derivatives. Derivatives are financial instruments that derive their value from an underlying asset (McDonald, 2013). Some popular examples of financial derivatives include futures, forwards, options, and swaps.

Bruno is a Julia (Bezanson et al., 2017) package that allows for comparison of different hedging and trading strategies, many of which are based on financial derivatives.

Statement of need

Bruno allows users to compare different financial derivatives, hedging, and trading strategies. One of the major benefits of Bruno is that it can calculate theoretical historical derivative prices for a variety of pricing models such as the Black-Scholes Model (Black & Scholes, 1973) and Monte Carlo Analysis (Clewlow & Strickland, 1998). This is important because derivative price data is not publicly available. Thus, many trading and hedging strategies, by necessity, are currently based on asset prices. Bruno allows them to be based on theoretical derivative prices instead.

Another key feature of Bruno is that it has the ability to produce a distribution of maximum loss that could result from a trading or hedging strategy. This information is valuable to financial practitioners and market makers as it helps quantify the risk of a potential strategy before putting the strategy into place. Bruno also allows for comparison of different trading and hedging strategies. Creation of these distributions is facilitated by Bruno's data generating processes. These processes include non-parametric methods, such as the stationary bootstrap (Politis & Romano, 1994) with automatic block-length selection (Patton et al., 2009; Politis & White, 2004) as well as parametric methods, such as log diffusion.

Bruno was designed to be used by finance professionals and academics alike. Financial analysis of trading and hedging strategies can be intensive. This package is designed to make this type of investigation more straightforward and accessible. Many other software packages can calculate derivative prices, simulate hedging, and generate data. For example, in the Julia programing language, FinancialDerivatives.jl (Brilhante, 2018), FinancialMonteCarlo.jl (Scaramuzzino, 2018), and Strategems.jl (Amos, 2019) are packages that are used for derivative asset pricing, data simulation, and strategy testing, respectively. However, none of these packages have been compiled in a manner that allows for integrated analysis. Each of the listed packages performs one part of the process independently and must be assembled by the

programmer. Bruno, on the other hand, is novel because it provides a replacement for these independent packages with a fully integrated set of tools for derivatives analysis designed to work in a unified manner. Bruno was recently used in a conference publication (Pound, 2022), with several other publications nearing completion.

Example usage

Defining a strategy

Here we demonstrate how to define and use a trading strategy for testing. In this example, a strategy is defined where a derivative asset and its underlying stock is bought every Friday (assuming a 5-day trading week) and held until the end of the month. The `buy` and `sell` functions are provided by Bruno to make defining a strategy easier.

```
using Bruno

# creating a new strategy subtype for dispatch
primitive type ExampleStrategy <: Hedging 8 end


# define the new strategy
import Bruno: buy, sell, strategy
function Bruno.strategy(fin_obj,
                pricing_model,
                strategy_mode::Type{ExampleStrategy},
                holdings,
                step;
                kwargs...)

    if step % 5 == 0
        # buy one FinancialInstrument every 5 days with no transaction costs
        buy(fin_obj, 1, holdings, pricing_model, 0)
        # buy one Stock every 5 days
        buy(fin_obj.widget, 1, holdings, pricing_model, 0)
    end
    return holdings
end
```

Setting up assets and running the strategy

Using the type system for derivatives assets in Bruno, we define a stock and a European call option as example assets to be used in the strategy. We then run the strategy on simulated data from a log diffusion model. In this example, the European stock option is priced using the Black Scholes Model. It is important to note that alternative pricing and data simulation models could be used simply by changing the types used. This means strategies can be analyzed using a variety of assumptions about the asset and market conditions.

All logic for interest accrued and transaction costs during the time steps are handled by the simulation environment. The code returns the cumulative return from the simulated strategy as well as the agent's holdings in the agent's portfolio at each timestep during the simulation. Thus, for more complicated strategies such as those that depend on the derivative or the underlying asset, the holdings can be analyzed using common statistical time series tools.

```
# create a random array to act as historic prices
historic_prices = rand(50.0:75.0, 40)
```

```
# create stock from daily 'historic' prices
stock = Stock(;
    prices=historic_prices,
    name="example_stock",
    timesteps_per_period = 252
)

# create European stock call option using the defined stock
option = EuroCallOption(stock, 60.0)

# create vector of simulated future prices using the log diffusion model
input = LogDiffInput(;
    nTimeStep=252,
    initial=50,
    volatility=.3,
    drift=.08
)
future_prices = vec(makedata(input, 1))

# run the strategy for 20 days assuming all prices are daily
cumulative_returns, holdings = strategy_returns(
    option,
    BlackScholes,
    ExampleStrategy,
    future_prices,
    20, # number of days (20) for the simulation to run
    252 # Assuming 252 trading days in a year
)
```

Acknowledgements

We acknowledge support from Analytics Solutions Center at the department of Data Analytics and Information Systems (DAIS) at Utah State University, Huntsman School of Business.

References

Amos, J. (2019). Strategems.jl. In *Github repository*. Github. https://github.com/dysonance/Strategems.jl

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654. https://doi.org/10.1086/260062

Brilhante, A. (2018). FinancialDerivatives.jl. In *Github repository*. Github. https://github.com/JuliaQuant/FinancialDerivatives.jl

Clewlow, L., & Strickland, C. (1998). *Implementing derivative models*. John Wiley & Sons Incorporated. https://doi.org/10.1057/9780230392687.0018

Culp, C. L. (2011). *Risk transfer: Derivatives in theory and practice*. John Wiley & Sons.

McDonald, R. L. (2013). *Derivatives markets*. Pearson. ISBN: 9780321543080

Patton, A., Politis, D. N., & White, H. (2009). Correction to "automatic block-length selection for the dependent bootstrap" by d. Politis and h. white. *Econometric Reviews*, *28*(4),

372–375. https://doi.org/10.1080/07474930802459016

Politis, D. N., & Romano, J. P. (1994). The stationary bootstrap. *Journal of the American Statistical Association*, *89*(428), 1303–1313. https://doi.org/10.1080/01621459.1994.10476870

Politis, D. N., & White, H. (2004). Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, *23*(1), 53–70. https://doi.org/10.1081/etc-120028836

Pound, M. (2022). *Black scholes delta hedge in imperfect markets*. https://symposium.foragerone.com/2022-usu-fall-student-research-symposium/presentations/50139.

Scaramuzzino, N. (2018). FinancialMonteCarlo.jl. In *Github repository*. Github. https://github.com/rcalxrc08/FinancialMonteCarlo.jl