

Simulación de un Bot de Trading usando Eventos Discretos

Darío López Falcón
Universidad de la Habana (UH)
Facultad de Matemática y Computación (MATCOM)

13 de abril de 2025

S1. Introducción

Este proyecto tiene como objetivo simular y analizar el comportamiento de un bot de trading usando técnicas de simulación de eventos discretos. Se busca evaluar cómo diferentes configuraciones del bot responden bajo distintos escenarios de mercado simulados estocásticamente. A partir de esto, se pretende aplicar herramientas estadísticas reales para realizar comparaciones, validar hipótesis y explorar el impacto de variables como la volatilidad del mercado, el horizonte temporal y la estrategia del bot.

Variables de interés: ganancia neta, cantidad de operaciones realizadas, valor final del portafolio, umbrales de compra y venta, comisión por transacción.

S2. Detalles de Implementación

Simulación del sistema

La simulación se implementó en Python, usando una arquitectura modular.

Estructura modular del código:

- `main.py`: ejemplo simple de simulación, sin hacer ningún estudio estadístico de los datos
- `market.py`: contiene la lógica para la generación de precios del activo.
- `bot.py`: clase base de los bots con estrategias basadas en umbrales.
- `simulator.py`: ejecución de la simulación.
- `analysis.py`: lógica de simulaciones múltiples, tests, bootstrap, paradas y más.
- `presentacion.ipynb`: presentación que agrupa varios análisis y estudios aplicados a distintas simulaciones

La lógica del bot está basada en umbrales: compra cuando el precio baja cierto porcentaje con respecto al anterior, y vende cuando supera un porcentaje de ganancia esperado. Se considera también una comisión por operación.

Modelos de generación de precios

Durante el desarrollo del proyecto, se implementaron y compararon varios modelos estocásticos para la simulación del comportamiento del precio de un activo financiero. A continuación, se detallan sus expresiones matemáticas:

- **Ruido Gaussiano Aditivo (sin tendencia):**

$$P_t = P_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- **Gaussiano con tendencia:**

$$P_t = P_{t-1} + \mu + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

donde μ representa una tendencia positiva o negativa en el mercado.

- **Movimiento Browniano Geométrico:**

$$P_t = P_{t-1} \cdot e^{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}Z_t}, \quad Z_t \sim \mathcal{N}(0, 1)$$

Este modelo es multiplicativo y asegura que los precios se mantengan positivos. Es ampliamente usado en finanzas para modelar precios de acciones.

- **Proceso Mean-Reverting (Ornstein-Uhlenbeck discreto):**

$$P_t = \theta + \phi(P_{t-1} - \theta) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

donde θ es el valor al que el precio tiende a regresar, y $0 < \phi < 1$ controla la velocidad de reversión.

Eventos simulados: cada nuevo precio genera un evento. Donde las decisiones de los bots se modelan como funciones deterministas sobre el estado actual:

$$\text{accion}(P_t) = \begin{cases} \text{comprar, si } P_t < P_{\text{compra}} \\ \text{vender, si } P_t > P_{\text{venta}} \\ \text{nada, eoc} \end{cases}$$

Bots definidos: se definieron múltiples bots con distintos umbrales de compra y venta, y comisiones por transacción. Las estrategias se mantienen simples para facilitar el análisis.

Herramientas estadísticas

Se desarrollaron funciones para:

- Simulación múltiple con distintas semillas.
- Estimación de intervalos de confianza.
- Parada estadística automática basada en error.
- Bootstrap para media y mediana.
- Pruebas de hipótesis entre bots y modelos de precios.

Visualización

Se generaron gráficas para:

- Decisiones de compra y venta.
- Evolución del portafolio y capital invertido.
- Distribuciones de resultados por estrategia, modelo y duración.

S3. Experimentos

Distribuciones de resultados

Se realizaron multiples simulaciones por configuración. La ganancia final y número de operaciones mostraron diferencias importantes entre bots.

Bootstrap e intervalo de confianza

Se aplicó bootstrap para estimar la media y el intervalo de confianza al 95 %. Esto permitió validar los resultados incluso cuando la distribución no era normal (según Shapiro y D'Agostino).

Criterio de parada

Se utilizó un umbral de error de \$100 para detener las simulaciones una vez alcanzado un nivel de confianza deseado.

Comparación de horizontes de tiempo

Se simuló el mismo bot con series de precios de diferente duración (semana, mes y año) para analizar cómo afecta el horizonte temporal al desempeño. Se observó que a mayor duración, el bot tiende a obtener mayor ganancia, aunque también aumenta la varianza.

Comparación de modelos de precios

Se evaluó el rendimiento del bot bajo distintos modelos estocásticos de precios: ruido gaussiano, tendencia positiva, browniano geométrico y mean-reverting. El modelo con tendencia permitió mayores ganancias, mientras que el modelo mean-reverting resultó en un desempeño más conservador.

S4. Modelo Matemático

Como se ha explicado anteriormente, el sistema simulado consiste en un bot de trading que toma decisiones de compra y venta de un activo financiero a lo largo del tiempo, según una estrategia basada en umbrales. El comportamiento del precio del activo es modelado como un proceso estocástico.

Estado del sistema

El estado del sistema en un instante de tiempo t está determinado por las siguientes variables:

- P_t : precio del activo en el tiempo t .
- C_t : capital disponible (efectivo) del bot.
- Q_t : cantidad de activos en posesión (posición abierta).
- V_t : valor total del portafolio, dado por:

$$V_t = C_t + Q_t \cdot P_t$$

Dinámica de decisiones

La política de decisiones del bot es discreta y se basa en comparaciones con el último precio de compra P_{buy} :

- **Comprar:** si $Q_t = 0$ y $P_t < P_{\text{buy}} \cdot \delta_b$, donde $\delta_b < 1$ es el umbral de compra.
- **Vender:** si $Q_t > 0$ y $P_t > P_{\text{buy}} \cdot \delta_s$, donde $\delta_s > 1$ es el umbral de venta.

Cada operación incluye un costo proporcional de comisión f (fee), por lo que las transacciones afectan el capital de la siguiente forma:

- **Compra:**

$$C_t \leftarrow C_t - q \cdot P_t \cdot (1 + f)$$

- **Venta:**

$$C_t \leftarrow C_t + Q_t \cdot P_t \cdot (1 - f), \quad Q_t \leftarrow 0$$

Generación de precios

El precio del activo evoluciona según un modelo estocástico, por ejemplo:

- Ruido gaussiano:

$$P_t = P_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- Con tendencia:

$$P_t = P_{t-1} + \mu + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- Movimiento browniano geométrico:

$$P_t = P_{t-1} \cdot e^{(\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}Z_t}, \quad Z_t \sim \mathcal{N}(0, 1)$$

- Mean-reverting:

$$P_t = \theta + \phi(P_{t-1} - \theta) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

Salida del sistema

El resultado de la simulación es el valor final del portafolio al tiempo T , es decir:

$$V_T = C_T + Q_T \cdot P_T$$

donde:

- V_t : valor total del bot en el tiempo t
- C_t : capital disponible en efectivo
- P_t : precio del activo en t
- Q_t : cantidad de activos en cartera

S5. Conclusiones

El desarrollo de este proyecto permitió modelar y analizar el desempeño de un bot de trading mediante simulación de eventos discretos. A lo largo del trabajo se estudiaron distintas estrategias, modelos de mercado y horizontes temporales. A continuación, se detallan los principales resultados y hallazgos:

- **Comparación de estrategias:** Se compararon dos bots con estrategias distintas:
 - *Bot Conservador* (umbral de compra 0.98, venta 1.02) tuvo una ganancia promedio de \$9,783 con 11 operaciones promedio por simulación.
 - *Bot Agresivo* (compra 0.95, venta 1.05) obtuvo una media de \$10,110 con 22 operaciones en promedio.

Las pruebas de Mann-Whitney U y KS mostraron que las diferencias eran estadísticamente significativas ($p < 0,05$).

- **Bootstrap e intervalos de confianza:** Para el Bot Agresivo, se estimó mediante bootstrap:

$$\text{Ganancia media bootstrap} = \$10,108 \quad IC_{95\%} = (\$10,030, \$10,190)$$

Lo cual confirma la estabilidad de la estrategia con una varianza moderada.

- **Modelos de precios:** Se compararon cuatro generadores:
 - *Ruido Gaussiano*: ganancia media \$9,850
 - *Tendencia Positiva*: \$10,260
 - *Geom. Browniano*: \$10,080
 - *Mean-Reverting*: \$9,700

El modelo con tendencia favoreció claramente al bot, elevando su desempeño en más de un 4%.

- **Horizontes temporales:** Se simuló el mismo bot con 3 duraciones distintas:

- *Semana (50 pasos)*: ganancia media \$9,580; trades: 6
- *Mes (200 pasos)*: ganancia media \$9,910; trades: 14
- *Año (1000 pasos)*: ganancia media \$10,320; trades: 36

A mayor horizonte, se observó una mejora en los resultados, aunque también aumentó la dispersión.

- **Criterio de parada:** Se aplicó un umbral de error de \$100 en la estimación de la media. El bot agresivo requirió 76 simulaciones para alcanzar este margen de error, optimizando recursos sin perder precisión.
- **Evolución del sistema:** Se graficó el valor del portafolio en el tiempo, el número de posiciones abiertas y el porcentaje del capital invertido. Estos análisis mostraron que el bot agresivo mantenía posiciones abiertas por más tiempo y con mayor frecuencia, aprovechando mejor las oportunidades del mercado simulado.
- **Distribuciones no normales:** En casi todos los casos, las pruebas de normalidad (Shapiro-Wilk y D'Agostino) arrojaron $p < 0,05$, por lo que se optó por pruebas no paramétricas en las comparaciones.

En resumen, se logró no solo modelar un sistema de trading como un sistema discreto, sino también aplicar técnicas de análisis estadístico robusto que permitieron evaluar, validar y comparar estrategias de forma rigurosa. El trabajo demuestra que incluso estrategias simples pueden generar valor si se ajustan al comportamiento del mercado, y que la simulación es una herramienta poderosa para explorar dicho espacio de decisiones.