Finanzas

AUTHOR
Eduardo Selim Martínez Mayorga

PUBLISHED
May 10, 2022

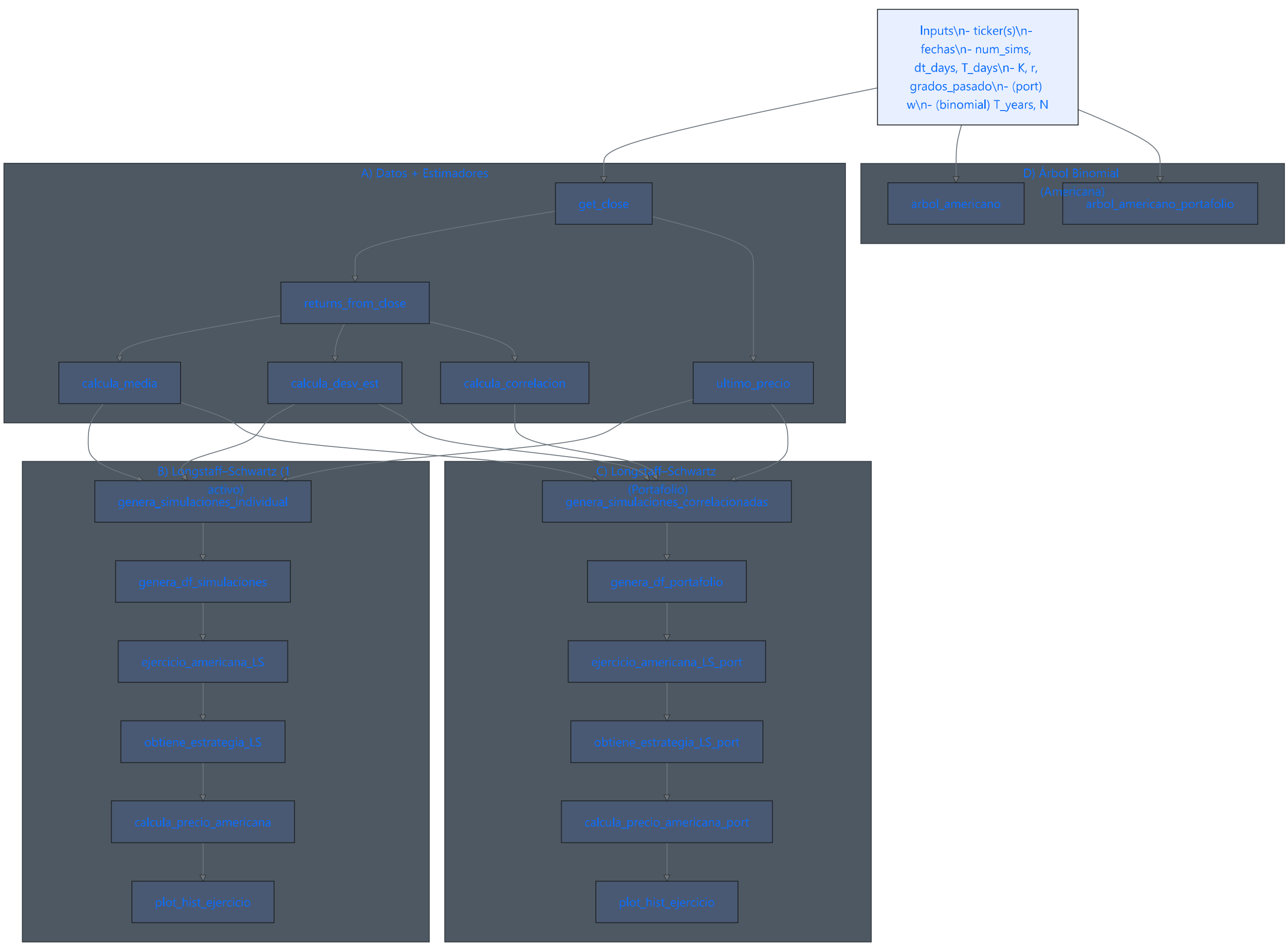
Finanzas

Proyecto individual

Profesor: Eduardo Selim Martínez Mayorga

Diagrama de flujo del notebook (Quarto)

> Da click en una función para ir a su ficha (inputs → proceso → outputs).



Librerías

▼ Code

```
import numpy as np
import pandas as pd
import yfinance as yf
import math
from functools import lru_cache
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

▼ Code

```
TRADING_DAYS = 252  # días hábiles por año
```

Descargar datos de Yahoo Finance

get_close

Recibe: `tickers`, `fecha_inicio`, `fecha_fin`
Hace: descarga precios de Yahoo Finance y devuelve un `DataFrame` con columnas=tickers.

▶ Code

returns_from_close

Recibe: `close: DataFrame`
Hace: calcula retornos log o simples.
Devuelve: `DataFrame` de retornos.

▶ Code

Estimadores

ultimo_precio

Recibe: - `ticker` (`str`): símbolo (ej. `"AAPL"`) - `fecha_inicio` (`str`): fecha inicio `"YYYY-MM-DD"` - `fecha_fin` (`str`): fecha fin `"YYYY-MM-DD"`

Qué hace: - Descarga la serie de precios de cierre (`Close` o `Adj Close`, según `get_close`) del ticker en el rango de fechas. - Toma el **último valor disponible** de la columna del ticker.

Devuelve: - `float` con el último precio observado: S_0.

▶ Code

calcula_media

Recibe: - ticker (str): símbolo (ej. "AAPL") - `fecha_inicio` (`str`): fecha inicio `"YYYY-MM-DD"` - `fecha_fin` (`str`): fecha fin `"YYYY-MM-DD"` - `kind` (str, default "log"): tipo de retorno: `log`: $r_t = \log(P_t) - \log(P_{t-1})$

Qué hace: - Descarga precios del activo en el rango. - Construye la serie de retornos diarios con `returns_from_close`. - Calcula la media muestral de esos retornos.

Devuelve: - `float` con la media diaria: $\mu_{\text{daily}} = \frac{1}{n}\sum_{t=1}^{n} r_t$.

▶ Code

calcula_desv_est

Recibe: - ticker (str): símbolo (ej. "AAPL") - `fecha_inicio` (`str`): fecha inicio `"YYYY-MM-DD"` - `fecha_fin` (`str`): fecha fin `"YYYY-MM-DD"` - `kind` (str, default "log"): tipo de retorno: `log`: $r_t = \log(P_t) - \log(P_{t-1})$ - `ddof` (int, default 1): grados de libertad para la desviación estándar: ddof=1 → muestral (recomendado para estimación) ddof=0 → "poblacional" (divide entre n)

Qué hace: - Descarga precios. - Calcula retornos diarios. - Calcula la desviación estándar de retornos (por default, muestral).

Devuelve: - `float` con la volatilidad diaria estimada: σ_{daily}.

▶ Code

calcula_correlacion

Recibe: - ticker (str): símbolo (ej. "AAPL") - `fecha_inicio` (`str`): fecha inicio `"YYYY-MM-DD"` - `fecha_fin` (`str`): fecha fin `"YYYY-MM-DD"` - `kind` (str, default "log"): tipo de retorno (log o simple)

Qué hace: - Descarga precios de ambos tickers. - Construye retornos diarios alineados por fecha (esto es importante: correlación requiere observaciones pareadas). - Calcula la correlación muestral de Pearson entre las dos series de retornos.

Devuelve: - `float` con la correlación: $\rho \in [-1, 1]$.

▶ Code

PAYOFFS (sin globals)

▶ Code

SIMULACIÓN GBM (1 activo) CON MU HISTÓRICA (P)

El modelo buscamos simular es el movimiento browniano geométrico (GBM):

$ dS_t = S_t,dt + S_t,dW_t. $

La discretización para un paso de tamaño Δt es:

$ S_{t+t} = S_t !((- ^2)t + ,Z), ZN(0,1). $

Por lo que tenemos dos factores indispensables: - El **drift** se multiplica por Δt. - El **término aleatorio** (browniano) se multiplica por $\sqrt{\Delta t}$.

En el código se construye W como una suma acumulada de normales estándar, $W = \sum Z$, y luego se usa $ S S_0(+ W) $, lo cual equivale a asumir implícitamente $\Delta t = 1$ (un paso de "tamaño 1") y además deja el parámetro `dt` sin afectar realmente la escala del proceso. Esto puede mezclar unidades (por ejemplo, interpretar $T = 252/2$ como días mientras el proceso usa pasos de tamaño 1 sin conversion).

La corrección consiste en introducir explícitamente el tamaño del paso: si el paso es de `dt_days` días y se trabaja en años, entonces $ t = . $ Con esto, la simulación correcta usa: $ S_{k+1} = S_k !((- ^2)t + ,Z_k). $

Así, al cambiar `dt_days` o el horizonte `T_days`, el modelo se escala correctamente y las unidades de μ, σ y el descuento (por ejemplo con $e^{-r\Delta t}$) quedan consistentes.

genera_simulaciones_individual

Simula GBM con $\Delta t = \frac{dt\_days}{252}$ y drift μ histórica.

▶ Code

LONGSTAFF–SCHWARTZ (1 activo)

ejercicio_americana_LS

Longstaff–Schwartz: backward induction + regresión para decidir ejercicio temprano (1 activo).

▶ Code

calcula_precio_americana

Calcula el precio como promedio del payoff descontado al tiempo de ejercicio.

▶ Code

SIMULACIÓN CORRELACIONADA (2 activos) CON MU HISTÓRICA (P)

genera_simulaciones_correlacionadas

Simula 2 GBM correlacionados vía Cholesky.

▶ Code

LONGSTAFF–SCHWARTZ (PORTAFOLIO)

ejercicio_americana_LS_port

Longstaff–Schwartz sobre portafolio $P = wS_1 + (1-w)S_2$.

▶ Code

calcula_precio_americana_port

Precio del derivado americano sobre el portafolio.

▶ Code

BINOMIAL AMERICANA (sola)

arbol_americano

Árbol binomial americano (1 activo).

▶ Code

arbol_americano_portafolio

Árbol binomial americano (portafolio con σ_P).

▶ Code

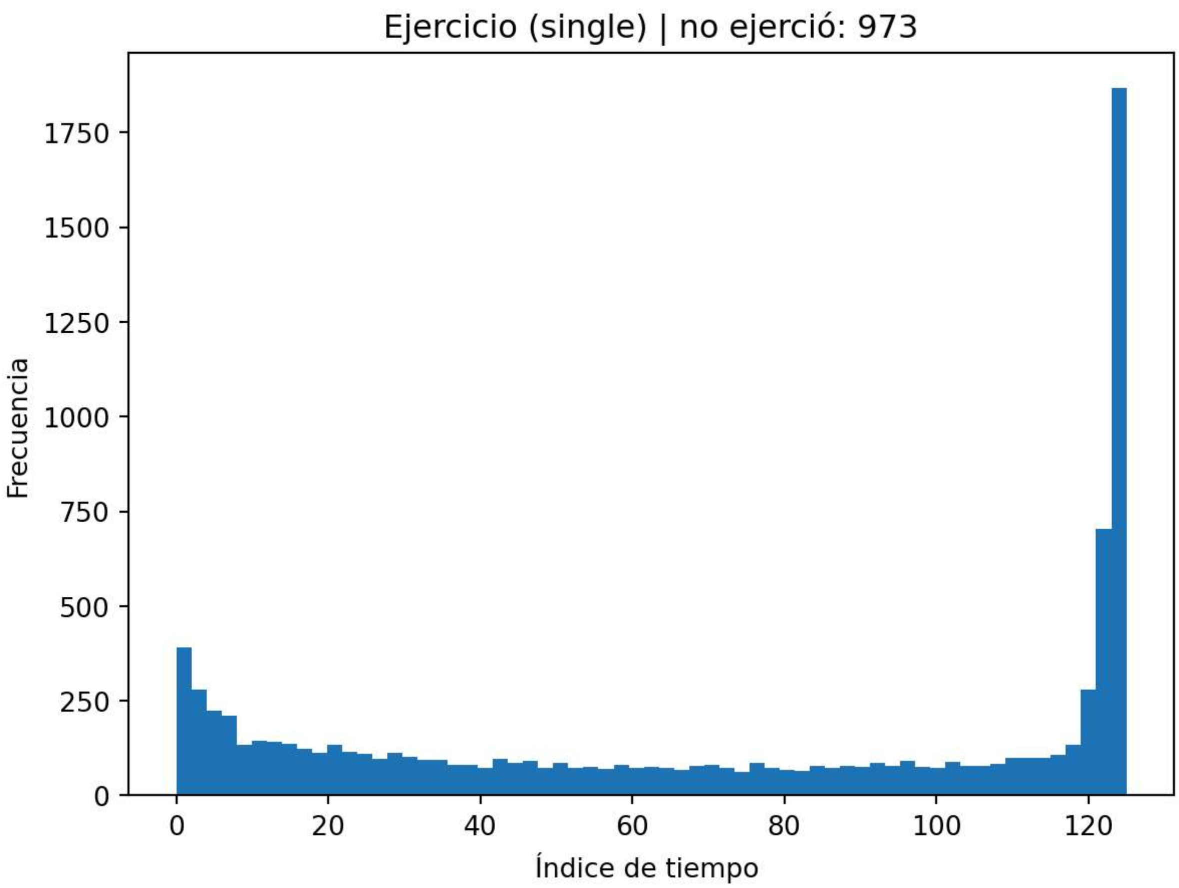
HISTOGRAMA

▶ Code

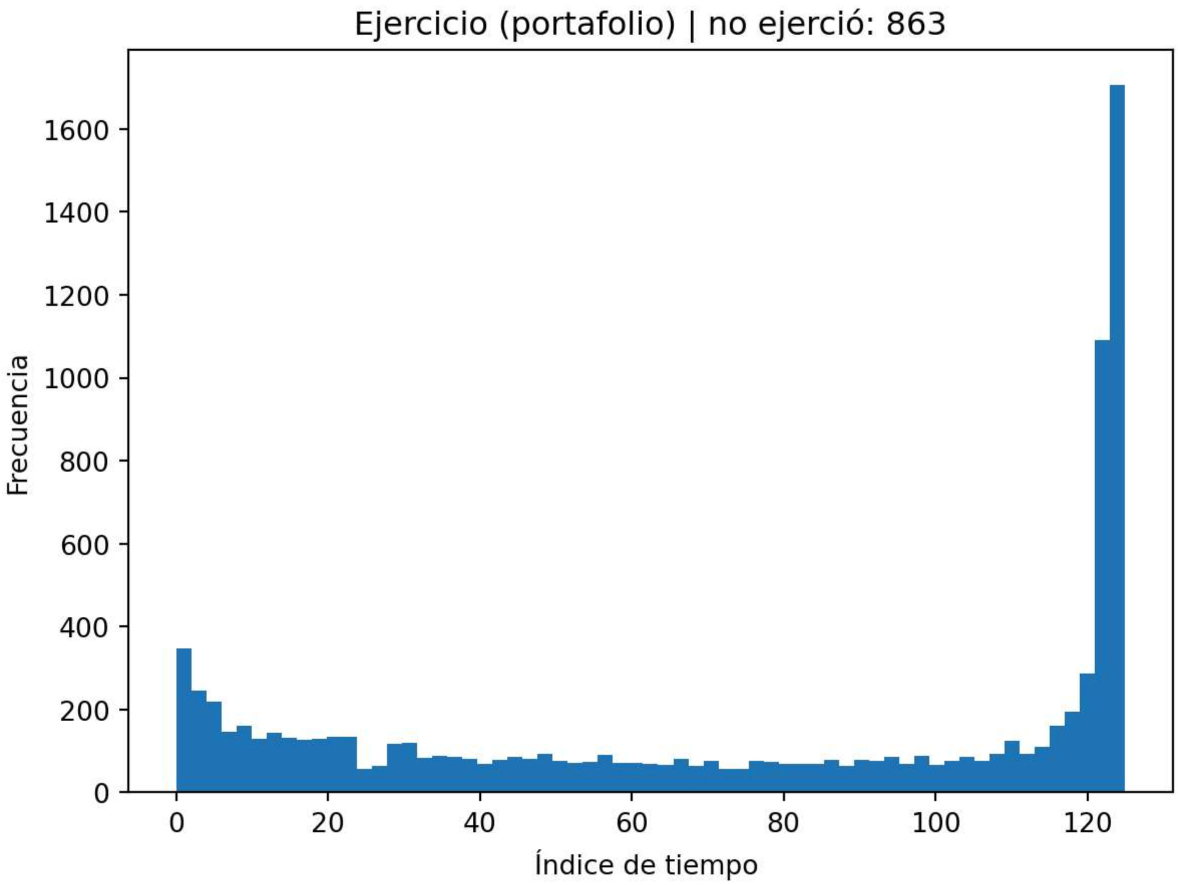
EJEMPLOS DE USO

▶ Code

```
Precio LS (single, put, con mu): 5.784524227975639
```




```
Precio LS (port, put, con mu): 5.024886585850465
```




```
Precio Binomial (single, call): 25.408046783678707
Precio Binomial (port, call): 29.09887840462954
```