untitled-3

October 1, 2023

[5]:
```python
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

[6]:
```python
# Define the cryptocurrency symbol and timeframe
crypto_symbol = 'BTC-USD'
start_date = '2020-01-01'
end_date = '2021-12-31'
```

[7]:
```python
# Download historical cryptocurrency price data using yfinance
crypto_data = yf.download(crypto_symbol, start=start_date, end=end_date)
```

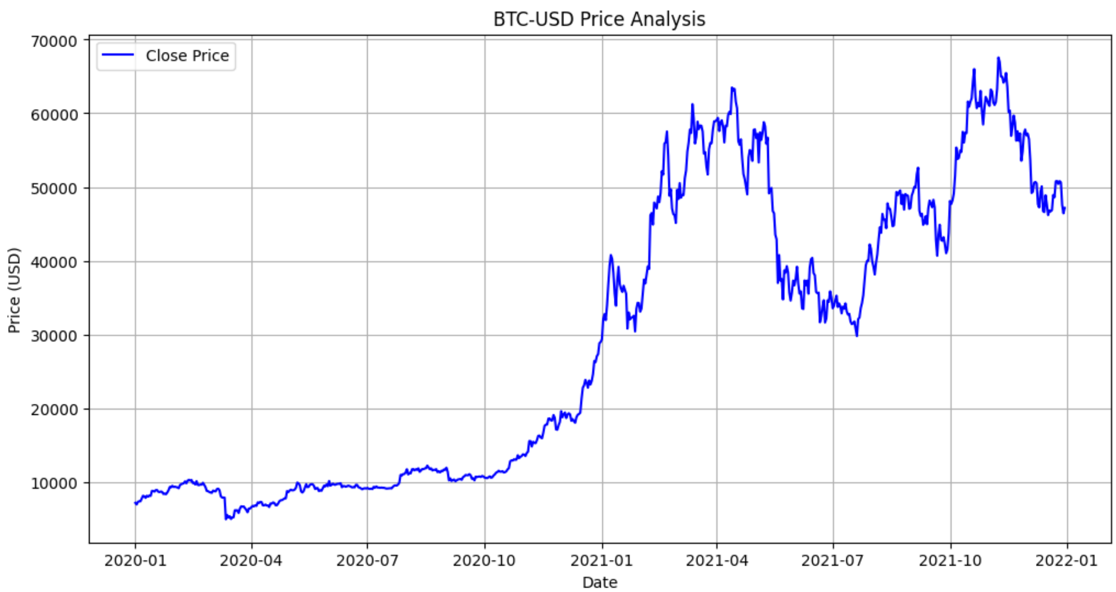
```
[*********************100%%**********************]  1 of 1 completed
```

[8]:
```python
# Display the first few rows of the data
print(crypto_data.head())
```

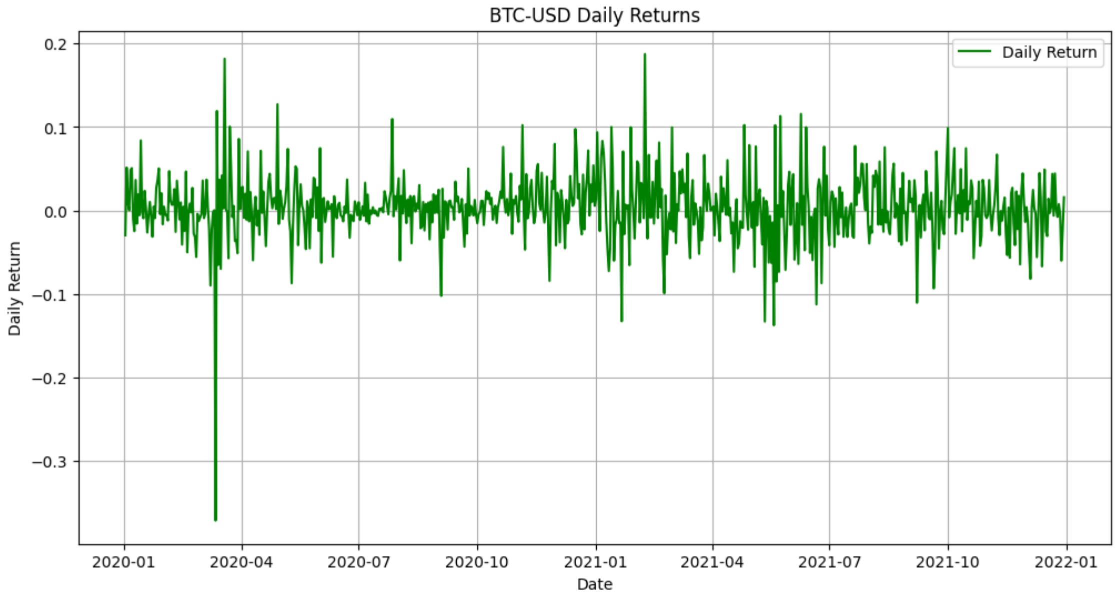
```
                   Open         High          Low        Close    Adj Close  \
Date
2020-01-01  7194.892090  7254.330566  7174.944336  7200.174316  7200.174316
2020-01-02  7202.551270  7212.155273  6935.270020  6985.470215  6985.470215
2020-01-03  6984.428711  7413.715332  6914.996094  7344.884277  7344.884277
2020-01-04  7345.375488  7427.385742  7309.514160  7410.656738  7410.656738
2020-01-05  7410.451660  7544.497070  7400.535645  7411.317383  7411.317383

                 Volume
Date
2020-01-01  18565664997
2020-01-02  20802083465
2020-01-03  28111481032
2020-01-04  18444271275
2020-01-05  19725074095
```

[9]:
```python
# Plotting the cryptocurrency price
plt.figure(figsize=(12, 6))
plt.title(f'{crypto_symbol} Price Analysis')
plt.plot(crypto_data['Close'], label='Close Price', color='blue')
```

```
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid()
plt.show()
```



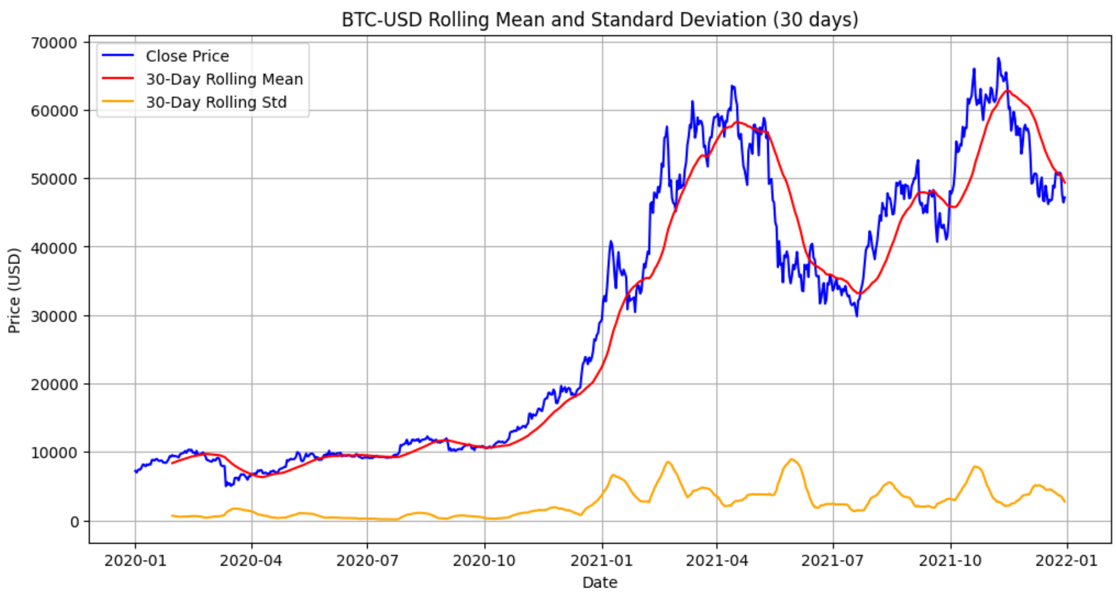
[48]:
```
# Calculate and visualize daily returns
crypto_data['Daily Return'] = crypto_data['Close'].pct_change()
plt.figure(figsize=(12, 6))
plt.title(f'{crypto_symbol} Daily Returns')
plt.plot(crypto_data['Daily Return'], label='Daily Return', color='green')
plt.xlabel('Date')
plt.ylabel('Daily Return')
plt.legend()
plt.grid()
plt.show()
```



[11]:
```python
# Calculate and visualize volatility
volatility = crypto_data['Daily Return'].std()
print(f'Volatility: {volatility:.4f}')
```

```
Volatility: 0.0400
```

[51]:
```python
# Calculate and visualize rolling mean and rolling standard deviation
rolling_mean = crypto_data['Close'].rolling(window=30).mean()
rolling_std = crypto_data['Close'].rolling(window=30).std()
plt.figure(figsize=(12, 6))
plt.title(f'{crypto_symbol} Rolling Mean and Standard Deviation (30 days)')
plt.plot(crypto_data['Close'], label='Close Price', color='blue')
plt.plot(rolling_mean, label='30-Day Rolling Mean', color='red')
plt.plot(rolling_std, label='30-Day Rolling Std', color='orange')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid()
plt.show()
```



[52]:
```
crypto_data['Daily Return'] = crypto_data['Adj Close'].pct_change()
```

[14]:
```
# Calculate daily mean return and standard deviation of return
mean_return = crypto_data['Daily Return'].mean()
std_return = crypto_data['Daily Return'].std()
```

[53]:
```
# Set the simulation parameters
initial_price = crypto_data['Adj Close'][-1]  # Current price
days = 365  # Number of days for the simulation
num_simulations = 1000  # Number of Monte Carlo simulation
```

[54]:
```
simulated_prices = pd.DataFrame()
```

[55]:
```
# Perform Monte Carlo simulations
for i in range(num_simulations):
    daily_returns = np.random.normal(mean_return, std_return, days)
    price_path = [initial_price]

    for j in range(1, days):
        price_path.append(price_path[j - 1] * (1 + daily_returns[j]))

    simulated_prices[f'Simulation {i + 1}'] = price_path
```

```
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

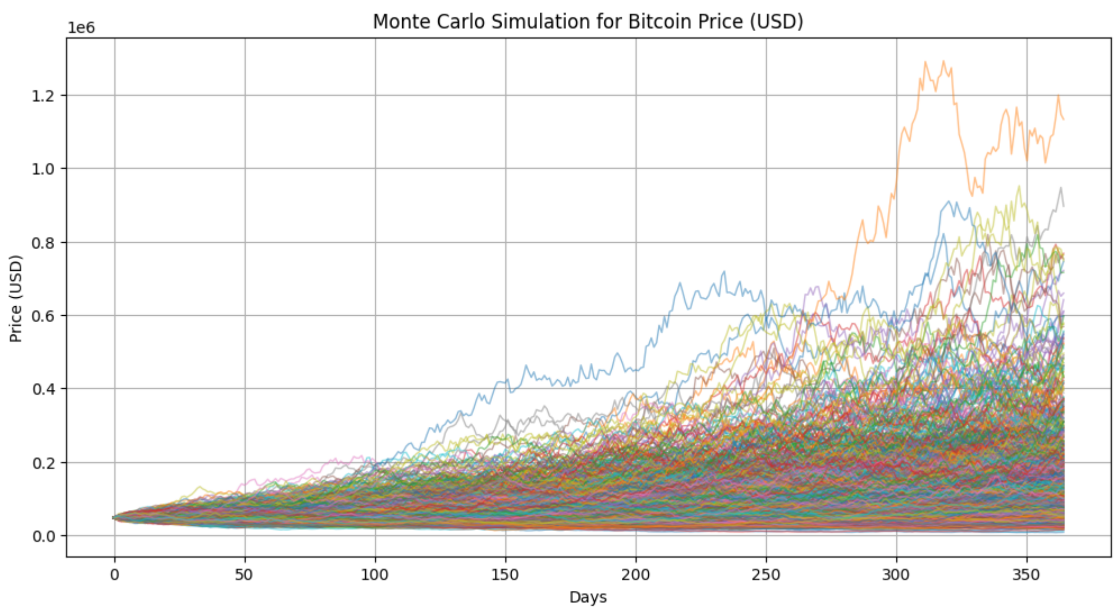
```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
```

```
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\648705944.py:9:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  simulated_prices[f'Simulation {i + 1}'] = price_path
```

[56]:
```python
# Plot the Monte Carlo simulation results
plt.figure(figsize=(12, 6))
plt.title('Monte Carlo Simulation for Bitcoin Price (USD)')
plt.xlabel('Days')
plt.ylabel('Price (USD)')
for i in range(num_simulations):
    plt.plot(simulated_prices[f'Simulation {i + 1}'], lw=1, alpha=0.5)
plt.grid()
plt.show()
```

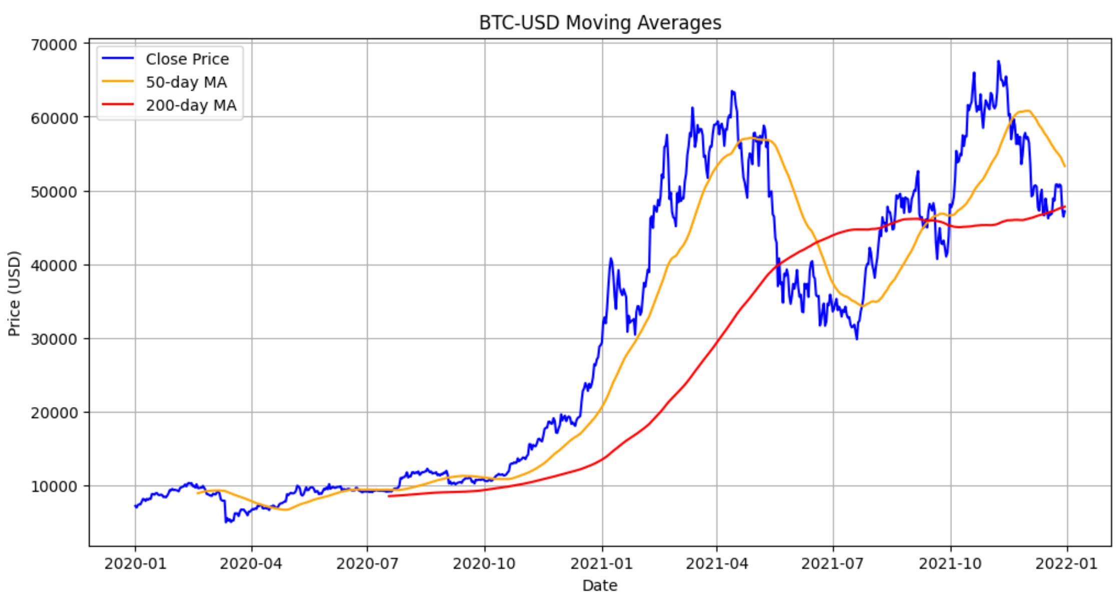


Technical Analysis

[21]:
```
# Moving Averages
short_window = 50  # Short-term moving average window
long_window = 200  # Long-term moving average window
```

[22]:
```
# Calculate short-term and long-term moving averages
crypto_data['Short_MA'] = crypto_data['Adj Close'].rolling(window=short_window).
 ↪mean()
crypto_data['Long_MA'] = crypto_data['Adj Close'].rolling(window=long_window).
 ↪mean()
```

[23]:
```
# Plot moving averages
plt.figure(figsize=(12, 6))
plt.title(f'{crypto_symbol} Moving Averages')
plt.plot(crypto_data.index, crypto_data['Adj Close'], label='Close Price',␣
 ↪color='blue')
plt.plot(crypto_data.index, crypto_data['Short_MA'], label=f'{short_window}-day␣
 ↪MA', color='orange')
plt.plot(crypto_data.index, crypto_data['Long_MA'], label=f'{long_window}-day␣
 ↪MA', color='red')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid()
plt.show()
```

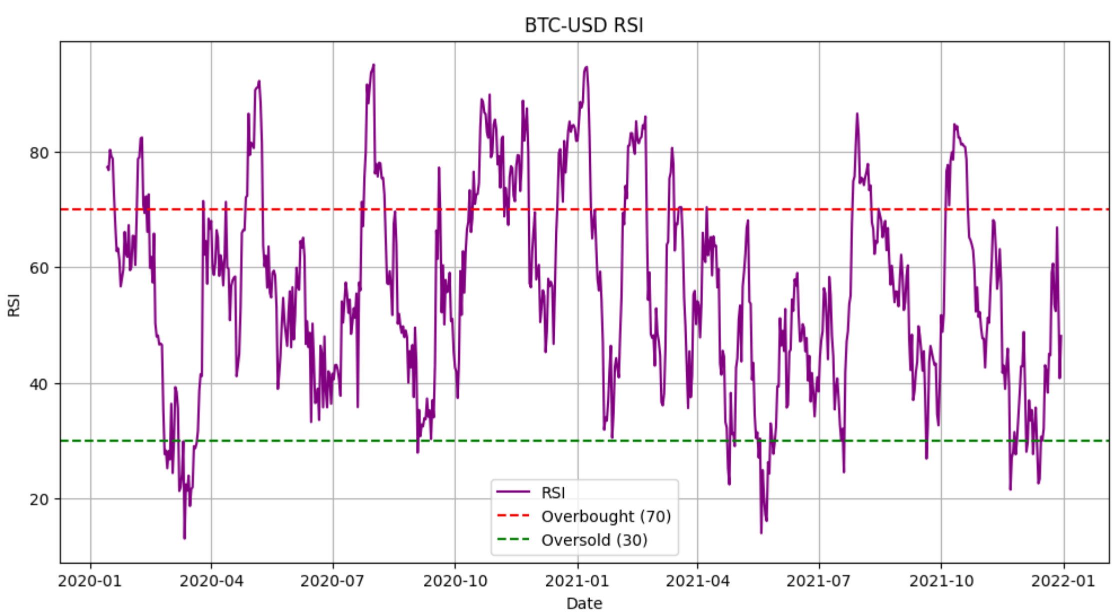


[24]:
```
# Relative Strength Index (RSI)
rsi_window = 14
```

[25]:
```
# Calculate RSI
delta = crypto_data['Adj Close'].diff(1)
gain = delta.where(delta > 0, 0)
loss = -delta.where(delta < 0, 0)
```

[26]:
```
avg_gain = gain.rolling(window=rsi_window).mean()
avg_loss = loss.rolling(window=rsi_window).mean()

rs = avg_gain / avg_loss
rsi = 100 - (100 / (1 + rs))
```

[27]:
```
# Plot RSI
plt.figure(figsize=(12, 6))
plt.title(f'{crypto_symbol} RSI')
plt.plot(crypto_data.index, rsi, label='RSI', color='purple')
plt.axhline(70, color='red', linestyle='--', label='Overbought (70)')
plt.axhline(30, color='green', linestyle='--', label='Oversold (30)')
plt.xlabel('Date')
plt.ylabel('RSI')
plt.legend()
plt.grid()
plt.show()
```



BACKTESTING

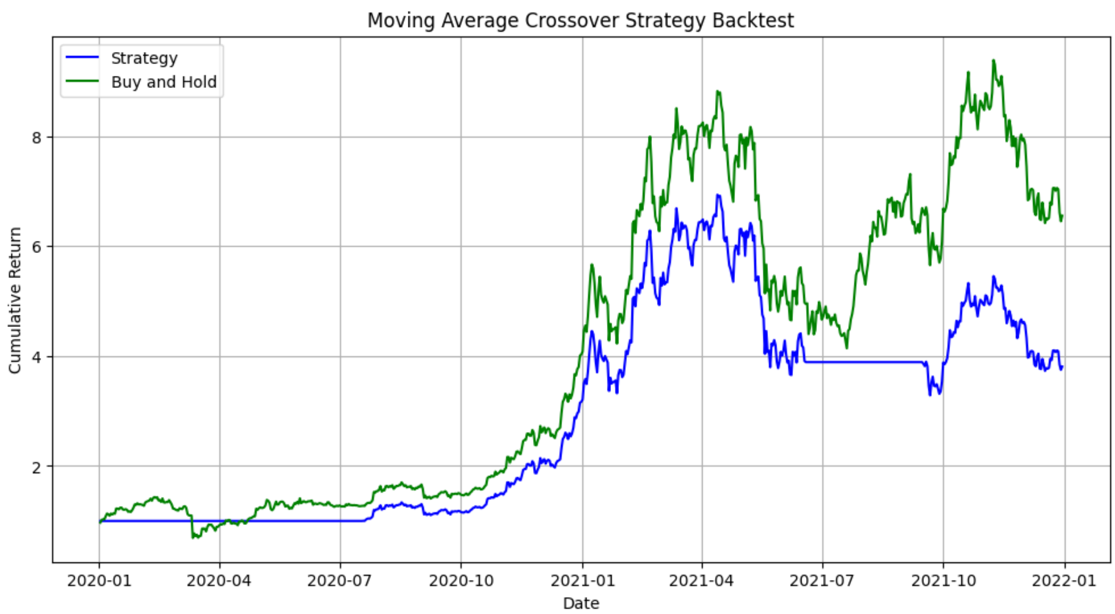
[35]:
```
# Create a column to signal buy/sell based on the moving average crossover
crypto_data['Signal'] = 0  # Initialize signal column
crypto_data['Signal'][short_window:] = np.
 ↪where(crypto_data['Short_MA'][short_window:] >␣
 ↪crypto_data['Long_MA'][short_window:], 1, 0)
```

```
C:\Users\akram\AppData\Local\Temp\ipykernel_26940\2833403269.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  crypto_data['Signal'][short_window:] =
np.where(crypto_data['Short_MA'][short_window:] >
crypto_data['Long_MA'][short_window:], 1, 0)
```

[36]:
```
# Calculate daily returns
crypto_data['Daily_Return'] = crypto_data['Adj Close'].pct_change()
crypto_data['Strategy_Return'] = crypto_data['Signal'].shift(1) *␣
 ↪crypto_data['Daily_Return']
```

[37]:
```
# Calculate cumulative returns
crypto_data['Cumulative_Strategy_Return'] = (1 +␣
 ↪crypto_data['Strategy_Return']).cumprod()
crypto_data['Cumulative_Holding_Return'] = (1 + crypto_data['Daily_Return']).
 ↪cumprod()
```

[38]:
```
# Plot cumulative returns
plt.figure(figsize=(12, 6))
plt.title('Moving Average Crossover Strategy Backtest')
plt.plot(crypto_data.index, crypto_data['Cumulative_Strategy_Return'],␣
 ↪label='Strategy', color='blue')
plt.plot(crypto_data.index, crypto_data['Cumulative_Holding_Return'],␣
 ↪label='Buy and Hold', color='green')
plt.xlabel('Date')
plt.ylabel('Cumulative Return')
plt.legend()
plt.grid()
plt.show()
```



[43]:
```
# Calculate performance metrics
strategy_cagr = (crypto_data['Cumulative_Strategy_Return'][-1] ** (252 /␣
↪len(crypto_data.index))) - 1  # Annualized CAGR
strategy_volatility = crypto_data['Strategy_Return'].std() * np.sqrt(252)  #␣
↪Annualized volatility
strategy_sharpe_ratio = strategy_cagr / strategy_volatility  # Sharpe ratio
strategy_max_drawdown = (crypto_data['Cumulative_Strategy_Return'] /␣
↪crypto_data['Cumulative_Strategy_Return'].cummax() - 1).min()  # Maximum␣
↪drawdown
```

[44]:
```
print(f"Strategy CAGR: {strategy_cagr:.2%}")
print(f"Strategy Volatility: {strategy_volatility:.2%}")
print(f"Strategy Sharpe Ratio: {strategy_sharpe_ratio:.2f}")
print(f"Strategy Maximum Drawdown: {strategy_max_drawdown:.2%}")
```

```
Strategy CAGR: 58.66%
Strategy Volatility: 48.38%
Strategy Sharpe Ratio: 1.21
Strategy Maximum Drawdown: -52.63%
```

[57]:
```
lags = 5
```

[59]:
```
cols = []
for lag in range(1, lags + 1):
    col = f'lag_{lag}'
    crypto_data[col] = crypto_data['Adj Close'].shift(lag)
    cols.append(col)
```

```
    crypto_data.dropna(inplace=True)
```

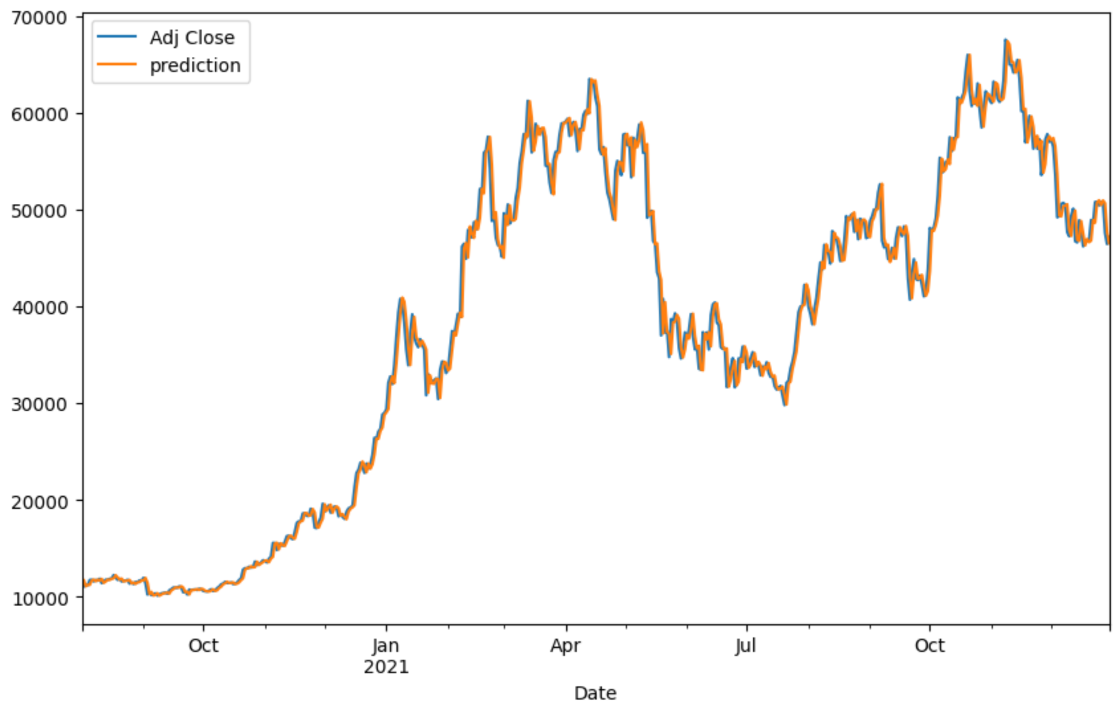
[60]:
```
reg = np.linalg.lstsq(crypto_data[cols], crypto_data['Adj Close'],
 rcond=None)[0]
```

[61]:
```
reg
```

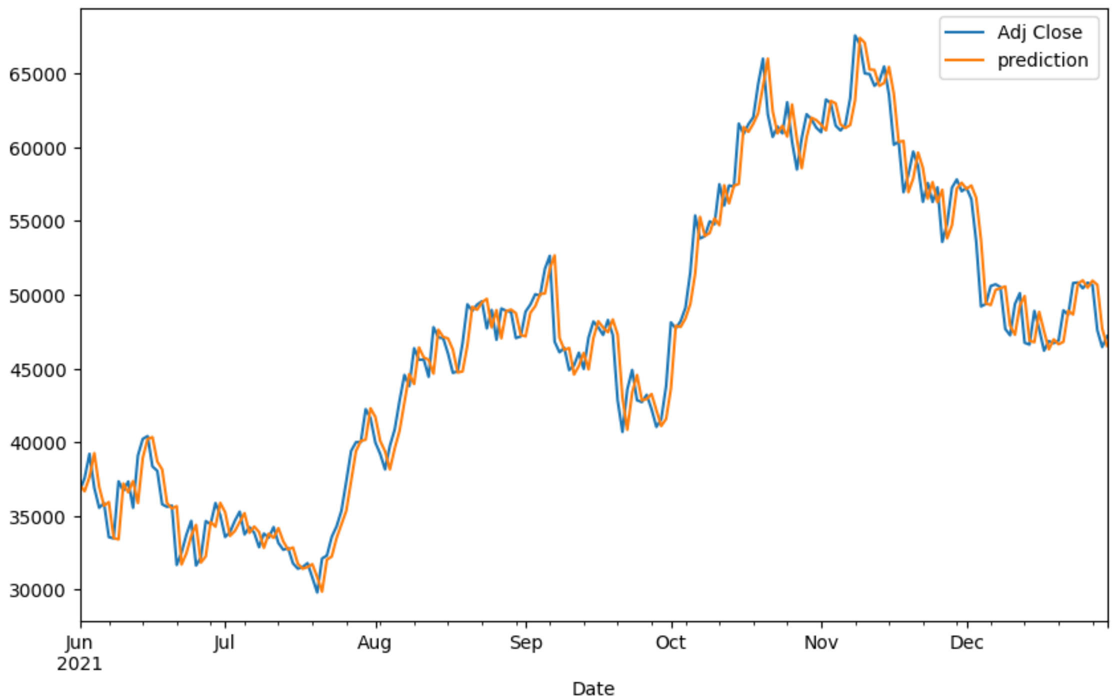
[61]:
```
array([ 0.9595198 ,  0.04205169,  0.00981083,  0.05248047, -0.06352994])
```

[63]:
```
crypto_data['prediction'] = np.dot(crypto_data[cols], reg)
```

[67]:
```
crypto_data[['Adj Close', 'prediction']].plot(figsize=(10, 6));
```



[68]:
```
crypto_data[['Adj Close', 'prediction']].loc['2021-06-1':].plot(
 figsize=(10, 6));
```



[74]:
```
crypto_data['return'] = np.log(crypto_data['Adj Close'] /
 crypto_data['Adj Close'].shift(1))
```

[75]:
```
crypto_data.dropna(inplace=True)
```

[78]:
```
cols = []
for lag in range(1, lags + 1):
     col = f'lag_{lag}'
     crypto_data[col] = crypto_data['return'].shift(lag)
     cols.append(col)
crypto_data.dropna(inplace=True)
```

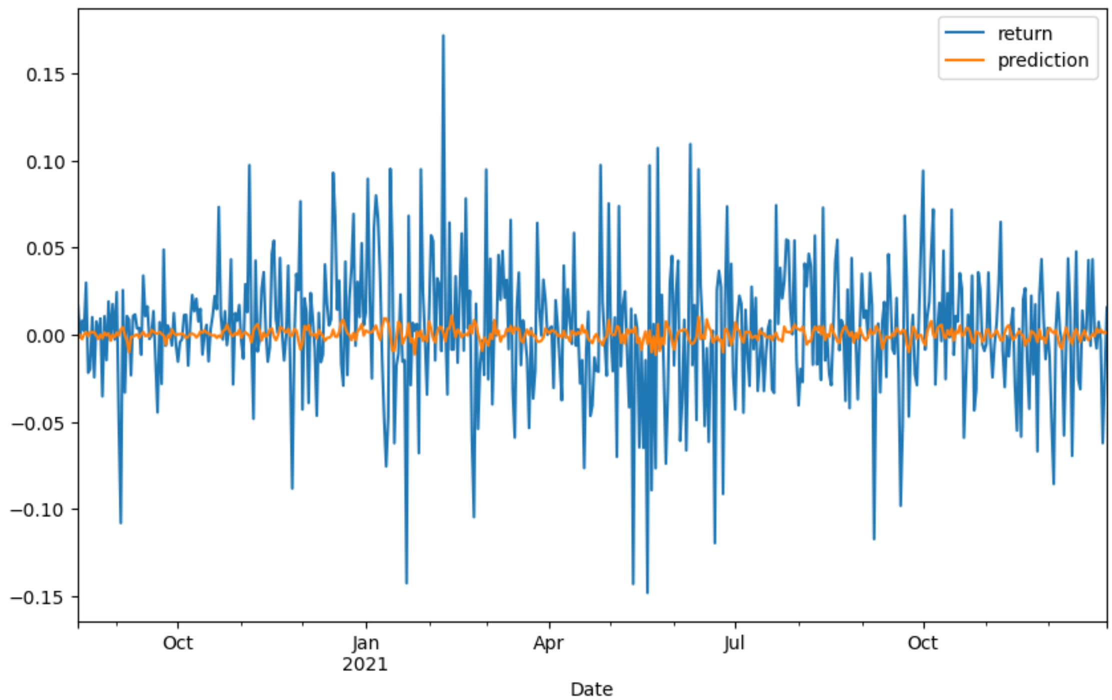
[80]:
```
reg = np.linalg.lstsq(crypto_data[cols], crypto_data['return'],
 rcond=None)[0]
```

[81]:
```
reg
```

[81]:
```
array([-0.03713645,  0.00523426,  0.01414599,  0.08030019,  0.02255332])
```

[82]:
```
crypto_data['prediction'] = np.dot(crypto_data[cols], reg)
```

[83]:
```
crypto_data[['return', 'prediction']].iloc[lags:].plot(figsize=(10, 6));
```




```
[86]: hits = np.sign(crypto_data['return'] *
       crypto_data['prediction']).value_counts()
```

```
[87]: hits
```

```
[87]:  1.0    268
      -1.0    242
      dtype: int64
```

```
[88]: hits.values[0] / sum(hits)
```

```
[88]: 0.5254901960784314
```

```
[89]: reg = np.linalg.lstsq(crypto_data[cols], np.sign(crypto_data['return']),
       rcond=None)[0]
```

```
[90]: reg
```

```
[90]: array([-1.66052253, -0.53561152,  0.80480457,  0.78128289,  0.28496472])
```

```
[92]: crypto_data['prediction'] = np.sign(np.dot(crypto_data[cols], reg))
```

```
[93]: crypto_data['prediction'].value_counts()
```

[93]:
```
 1.0    261
-1.0    249
Name: prediction, dtype: int64
```

[94]:
```
hits = np.sign(crypto_data['return'] *
 crypto_data['prediction']).value_counts()
```

[95]:
```
hits
```

[95]:
```
 1.0    270
-1.0    240
dtype: int64
```

[96]:
```
hits.values[0] / sum(hits)
```

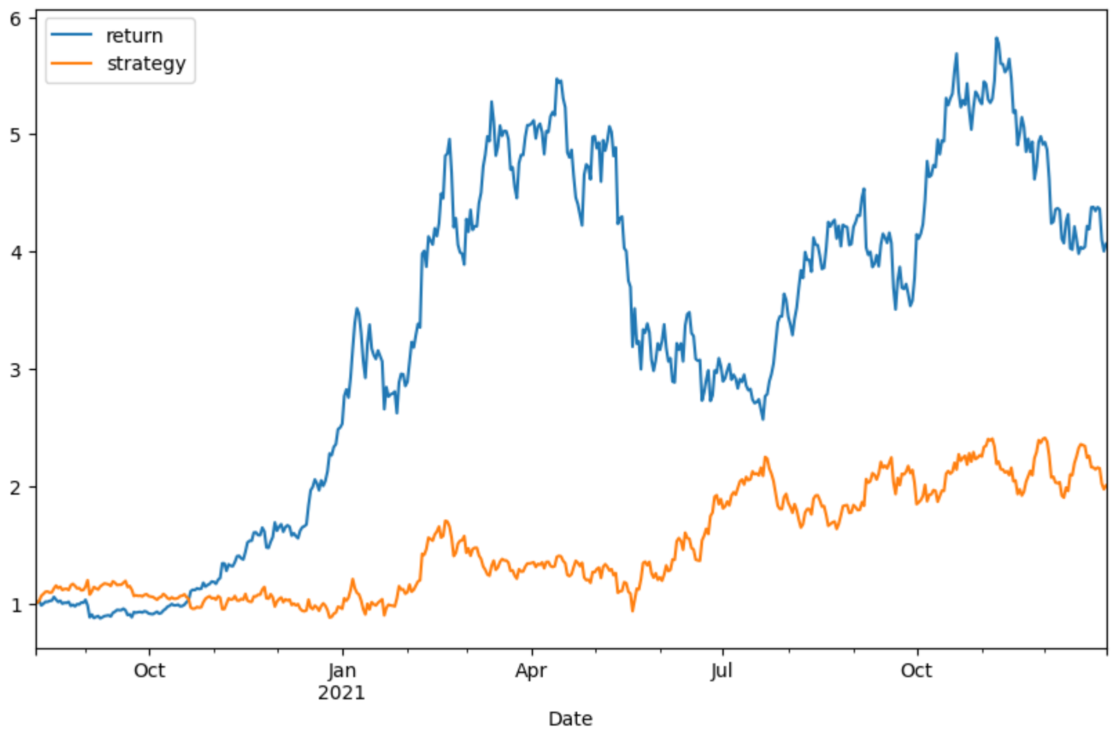
[96]:
```
0.5294117647058824
```

[97]:
```
crypto_data['strategy'] = crypto_data['prediction'] * crypto_data['return']
```

[98]:
```
crypto_data[['return', 'strategy']].sum().apply(np.exp)
```

[98]:
```
return      4.066563
strategy    2.005430
dtype: float64
```

[99]:
```
crypto_data[['return', 'strategy']].dropna().cumsum(
 ).apply(np.exp).plot(figsize=(10, 6));
```



return
strategy
6
5
4
3
2
1
Oct
Jan
2021
Apr
Jul
Oct
Date