MMF2000 RISK MANAGEMENT ASSIGNMENT 2

Market Risk and Counterparty Credit Risk (CCR)

Xinyi (Cynthia) Shen, 1005778428
xinyi.shen@mail.utoronto.ca

Due Date: November 14, 2024

1 P&L Attribution

Consider a European call option with the following features: strike price $K = \$105$ and the maturity is 1 year from today.

Yesterday, the stock price was \$100, the dividend yield was 4%, the risk-free rate was 2%, and volatility was 30%.

Today, the stock price is \$101, the dividend yield is 4%, the risk-free rate is 2.1%, and volatility is 32%.

Determine the P&L attribution by doing a sequence of valuations while changing the parameters in this order:

- Stock price
- Time decay
- Dividend yield
- Risk-free rate
- Volatility

Next work out the P&L attribution with the order reversed. Are the numbers the same? Why or why not?

Answer:

Case 1: Assume 365 days in a year

The Black-Scholes formula for a European call option price C is

$$C = V(S, K, r, q, \sigma, T) = Se^{-qT}\Phi(d_+) - e^{-rT}K\Phi(d_-),$$

where $\phi(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$, $\Phi(x) = \int_{-\infty}^{x} \phi(z)dz$, and $d_\pm = \frac{\ln\frac{S}{K} + (r - q \pm \frac{\sigma^2}{2}T)}{\sigma\sqrt{T}}$.

Yesterday, the European call option price C_0 was

$$\begin{aligned} C_0 &= V(S_1 = \$100, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1) \\ &= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% + \frac{30\%^2}{2})}{30\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% - \frac{30\%^2}{2})}{30\%}\right) \\ &= \$8.7510 \end{aligned}$$

Original Order:

When the *stock price* changes from \$100 to \$101, the European call option price changes from C_0 to C_S:

$$\begin{aligned} C_S &= V(S_2 = \$101, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1) \\ &= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% + \frac{30\%^2}{2})}{30\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% - \frac{30\%^2}{2})}{30\%}\right) \\ &= \$9.2074 \end{aligned}$$

The P&L attribution is

$$\delta_1^{origin} = C_S - C_0 = \$9.2074 - \$8.7510 = \$0.4564$$

When the *time decay* changes from 1 to $1 - \frac{1}{365}$, the European call option price changes from C_S to C_T:

$$\begin{aligned} C_T &= V(S_2 = \$101, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1 - \frac{1}{365}) \\ &= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% + \frac{30\%^2}{2}(1 - \frac{1}{365}))}{30\%\sqrt{1 - \frac{1}{365}}}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% - \frac{30\%^2}{2}(1 - \frac{1}{365}))}{30\%\sqrt{1 - \frac{1}{365}}}\right) \\ &= \$9.1946 \end{aligned}$$

The P&L attribution is

$$\delta_2^{origin} = C_S - C_0 = \$9.1946 - \$9.2074 = -\$0.0128$$

Since the *dividend yield* does not change, then we have $C_D = C_T = \$9.1946$ and $\delta_3^{origin} = \$0$.

When the *risk-free rate* changes from 2% to 2.1%, the European call option price changes from C_D to C_{R_f}:

$$\begin{aligned} C_{R_f} &= V(S_2 = \$101, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 30\%, T = 1 - \frac{1}{365}) \\ &= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{30\%^2}{2}(1 - \frac{1}{365}))}{30\%\sqrt{1 - \frac{1}{365}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{30\%^2}{2}(1 - \frac{1}{365}))}{30\%\sqrt{1 - \frac{1}{365}}}\right) \\ &= \$9.2320 \end{aligned}$$

The P&L attribution is

$$\delta_4^{origin} = C_D - C_{R_f} = \$9.2320 - \$9.1946 = \$0.0375$$

When the *volatility* changes from 30% to 32%, the European call option price changes from C_{R_f} to C_σ:

$$\begin{aligned} C_\sigma &= V(S_2 = \$101, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1 - \frac{1}{365}) \\ &= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) \\ &= \$10.0049 \end{aligned}$$

The P&L attribution is

$$\delta_5^{origin} = C_{R_f} - C_\sigma = \$10.0049 - \$9.2320 = \$0.7729$$

Reverse Order:

When the *volatility* changes from 30% to 32%, the European call option price changes from C_0 to C_σ:

$$\begin{aligned}
C_\sigma &= V(S_2 = \$100, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\
&= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% + \frac{32\%^2}{2})}{32\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% - \frac{32\%^2}{2})}{32\%}\right) \\
&= \$9.5159
\end{aligned}$$

The P&L attribution is

$$\delta_1^{reverse} = C_\sigma - C_0 = \$9.5159 - \$8.7510 = \$0.7649$$

When the *risk-free rate* changes from 2% to 2.1%, the European call option price changes from C_σ to C_{R_f}:

$$\begin{aligned}
C_{R_f} &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\
&= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2})}{32\%}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2})}{32\%}\right) \\
&= \$9.5523
\end{aligned}$$

The P&L attribution is

$$\delta_2^{reverse} = C_{R_f} - C_\sigma = \$9.5523 - \$9.5159 = \$0.0365$$

Since the *dividend yield* does not change, then we have $C_D = C_{R_f} = \$9.5523$ and $\delta_3^{reverse} = \$0$.

When the *time decay* changes from 1 to $1 - \frac{1}{365}$, the European call option price changes from C_D to C_T:

$$\begin{aligned}
C_T &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1 - \frac{1}{365}) \\
&= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) \\
&= \$9.5385
\end{aligned}$$

The P&L attribution is

$$\delta_4^{reverse} = C_T - C_D = \$9.5385 - \$9.5523 = -\$0.0138$$

When the *stock price* changes from \$100 to \$101, the European call option price changes from C_T to C_S:

$$\begin{aligned}
C_S &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\
&= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{365}))}{32\%\sqrt{1 - \frac{1}{365}}}\right) \\
&= \$10.0049
\end{aligned}$$

The P&L attribution is

$$\delta_5^{reverse} = C_S - C_T = \$10.0049 - \$9.5385 = \$0.4664$$

Python Code and Output:

```python
from scipy.stats import norm
import numpy as np

# Define the Black-Scholes call option price function
def black_scholes_call(S, K, T, r, sigma, q):
    d1 = (np.log(S / K) + (r - q + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    call_price = S * np.exp(-q * T) * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    return call_price

# Initial parameters
K = 105               # Strike price
initial_S = 100       # Initial stock price
final_S = 101         # Final stock price
q = 0.04              # Dividend yield
initial_r = 0.02      # Initial risk-free rate
final_r = 0.021       # Final risk-free rate
initial_sigma = 0.3   # Initial volatility
final_sigma = 0.32    # Final volatility
T = 1                 # Time to maturity in years
T_minus_day = T - (1/365)  # Adjust for one day's time decay

# Step 1: Calculate initial option price
initial_price = black_scholes_call(initial_S, K, T, initial_r, initial_sigma, q)
# Perform adjustments in the given order and record the incremental changes
# 1. Adjust stock price
price_after_stock = black_scholes_call(final_S, K, T, initial_r, initial_sigma, q)
delta_stock = price_after_stock - initial_price
# 2. Adjust for time decay (one day less)
price_after_time = black_scholes_call(final_S, K, T_minus_day, initial_r, initial_sigma, q)
delta_time = price_after_time - price_after_stock
# 3. Adjust dividend yield (stays the same, so no change expected)
price_after_dividend = black_scholes_call(final_S, K, T_minus_day, initial_r, initial_sigma, q)
delta_dividend = price_after_dividend - price_after_time
# 4. Adjust risk-free rate
price_after_rate = black_scholes_call(final_S, K, T_minus_day, final_r, initial_sigma, q)
delta_rate = price_after_rate - price_after_dividend
# 5. Adjust volatility
final_price = black_scholes_call(final_S, K, T_minus_day, final_r, final_sigma, q)
delta_volatility = final_price - price_after_rate
# Results in given order
order1_results = {
    'Initial Price': initial_price,
    'After Stock Price Change': price_after_stock,
    'Delta Stock Price': delta_stock,
    'After Time Decay': price_after_time,
    'Delta Time Decay': delta_time,
    'After Dividend Yield': price_after_dividend,
    'Delta Dividend Yield': delta_dividend,
    'After Risk-Free Rate Change': price_after_rate,
    'Delta Risk-Free Rate': delta_rate,
    'After Volatility Change (Final Price)': final_price,
    'Delta Volatility': delta_volatility,
}

# Perform calculations in the reverse order
# 1. Adjust volatility first
price_after_volatility_reverse = black_scholes_call(initial_S, K, T, initial_r, final_sigma, q)
delta_volatility_reverse = price_after_volatility_reverse - initial_price
```

```python
# 2. Adjust risk-free rate
price_after_rate_reverse = black_scholes_call(initial_S, K, T, final_r, final_sigma, q)
delta_rate_reverse = price_after_rate_reverse - price_after_volatility_reverse
# 3. Adjust dividend yield (stays the same, so no change expected)
price_after_dividend_reverse = black_scholes_call(initial_S, K, T, final_r, final_sigma, q)
delta_dividend_reverse = price_after_dividend_reverse - price_after_rate_reverse
# 4. Adjust for time decay
price_after_time_reverse = black_scholes_call(initial_S, K, T_minus_day, final_r, final_sigma, q)
delta_time_reverse = price_after_time_reverse - price_after_dividend_reverse
# 5. Adjust stock price
final_price_reverse = black_scholes_call(final_S, K, T_minus_day, final_r, final_sigma, q)
delta_stock_reverse = final_price_reverse - price_after_time_reverse
# Results in reverse order
order2_results = {
    'Initial Price': initial_price,
    'After Volatility Change': price_after_volatility_reverse,
    'Delta Volatility': delta_volatility_reverse,
    'After Risk-Free Rate Change': price_after_rate_reverse,
    'Delta Risk-Free Rate': delta_rate_reverse,
    'After Dividend Yield': price_after_dividend_reverse,
    'Delta Dividend Yield': delta_dividend_reverse,
    'After Time Decay': price_after_time_reverse,
    'Delta Time Decay': delta_time_reverse,
    'After Stock Price Change (Final Price)': final_price_reverse,
    'Delta Stock Price': delta_stock_reverse,
}
# Format the results to 4 decimal points and display
order1_results_rounded = {k: round(v, 4) for k, v in order1_results.items()}
order2_results_rounded = {k: round(v, 4) for k, v in order2_results.items()}
order1_results_rounded, order2_results_rounded, round(final_price, 4), round(final_price_reverse, 4)
```

```
({'Initial Price': 8.751,
  'After Stock Price Change': 9.2074,
  'Delta Stock Price': 0.4564,
  'After Time Decay': 9.1946,
  'Delta Time Decay': -0.0128,
  'After Dividend Yield': 9.1946,
  'Delta Dividend Yield': 0.0,
  'After Risk-Free Rate Change': 9.232,
  'Delta Risk-Free Rate': 0.0375,
  'After Volatility Change (Final Price)': 10.0049,
  'Delta Volatility': 0.7729},
 {'Initial Price': 8.751,
  'After Volatility Change': 9.5159,
  'Delta Volatility': 0.7649,
  'After Risk-Free Rate Change': 9.5523,
  'Delta Risk-Free Rate': 0.0365,
  'After Dividend Yield': 9.5523,
  'Delta Dividend Yield': 0.0,
  'After Time Decay': 9.5385,
  'Delta Time Decay': -0.0138,
  'After Stock Price Change (Final Price)': 10.0049,
  'Delta Stock Price': 0.4664},
 10.0049,
 10.0049)
```

Case 2: Assume 252 days in a year

Yesterday, the European call option price C_0 was

$$\begin{aligned}
C_0 &= V(S_1 = \$100, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1) \\
&= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% + \frac{30\%^2}{2})}{30\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% - \frac{30\%^2}{2})}{30\%}\right) \\
&= \$8.7510
\end{aligned}$$

Original Order:

When the *stock price* changes from \$100 to \$101, the European call option price changes from C_0 to C_S:

$$\begin{aligned}
C_S &= V(S_2 = \$101, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1) \\
&= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% + \frac{30\%^2}{2})}{30\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% - \frac{30\%^2}{2})}{30\%}\right) \\
&= \$9.2074
\end{aligned}$$

The P&L attribution is

$$\delta_1^{origin} = C_S - C_0 = \$9.2074 - \$8.7510 = \$0.4564$$

When the *time decay* changes from 1 to $1 - \frac{1}{252}$, the European call option price changes from C_S to C_T:

$$\begin{aligned}
C_T &= V(S_2 = \$101, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 30\%, T = 1 - \frac{1}{252}) \\
&= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% + \frac{30\%^2}{2}(1 - \frac{1}{252}))}{30\%\sqrt{1 - \frac{1}{252}}}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2\% - 4\% - \frac{30\%^2}{2}(1 - \frac{1}{252}))}{30\%\sqrt{1 - \frac{1}{252}}}\right) \\
&= \$9.1888
\end{aligned}$$

The P&L attribution is

$$\delta_2^{origin} = C_S - C_0 = \$9.1888 - \$9.2074 = -\$0.0186$$

Since the *dividend yield* does not change, then we have $C_D = C_T = \$9.1888$ and $\delta_3^{origin} = \$0$.

When the *risk-free rate* changes from 2% to 2.1%, the European call option price changes from C_D to C_{R_f}:

$$\begin{aligned}
C_{R_f} &= V(S_2 = \$101, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 30\%, T = 1 - \frac{1}{252}) \\
&= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{30\%^2}{2}(1 - \frac{1}{252}))}{30\%\sqrt{1 - \frac{1}{252}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{30\%^2}{2}(1 - \frac{1}{252}))}{30\%\sqrt{1 - \frac{1}{252}}}\right) \\
&= \$9.2262
\end{aligned}$$

The P&L attribution is

$$\delta_4^{origin} = C_D - C_{R_f} = \$9.2262 - \$9.1888 = \$0.0374$$

When the *volatility* changes from 30% to 32%, the European call option price changes from C_{R_f} to C_σ:

$$\begin{aligned}
C_\sigma &= V(S_2 = \$101, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1 - \frac{1}{252}) \\
&= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{252}))}{32\%\sqrt{1 - \frac{1}{252}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{252}))}{32\%\sqrt{1 - \frac{1}{252}}}\right) \\
&= \$9.9987
\end{aligned}$$

The P&L attribution is

$$\delta_5^{origin} = C_{R_f} - C_\sigma = \$9.9987 - \$9.2262 = \$0.7724$$

Reverse Order:

When the *volatility* changes from 30% to 32%, the European call option price changes from C_0 to C_σ:

$$\begin{aligned} C_\sigma &= V(S_2 = \$100, K_1 = \$105, r_1 = 2\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\ &= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% + \frac{32\%^2}{2})}{32\%}\right) - \$105e^{-2\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2\% - 4\% - \frac{32\%^2}{2})}{32\%}\right) \\ &= \$9.5159 \end{aligned}$$

The P&L attribution is

$$\delta_1^{reverse} = C_\sigma - C_0 = \$9.5159 - \$8.7510 = \$0.7649$$

When the *risk-free rate* changes from 2% to 2.1%, the European call option price changes from C_σ to C_{R_f}:

$$\begin{aligned} C_{R_f} &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\ &= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2})}{32\%}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2})}{32\%}\right) \\ &= \$9.5523 \end{aligned}$$

The P&L attribution is

$$\delta_2^{reverse} = C_{R_f} - C_\sigma = \$9.5523 - \$9.5159 = \$0.0365$$

Since the *dividend yield* does not change, then we have $C_D = C_{R_f} = \$9.5523$ and $\delta_3^{reverse} = \$0$.

When the *time decay* changes from 1 to $1 - \frac{1}{25}$, the European call option price changes from C_D to C_T:

$$\begin{aligned} C_T &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1 - \frac{1}{25}) \\ &= \$100e^{-4\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{25}))}{32\%\sqrt{1 - \frac{1}{25}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$100}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{25}))}{32\%\sqrt{1 - \frac{1}{25}}}\right) \\ &= \$9.5323 \end{aligned}$$

The P&L attribution is

$$\delta_4^{reverse} = C_T - C_D = \$9.5323 - \$9.5523 = -\$0.0201$$

When the *stock price* changes from \$100 to \$101, the European call option price changes from C_T to C_S:

$$\begin{aligned} C_S &= V(S_2 = \$100, K_1 = \$105, r_1 = 2.1\%, q_1 = 4\%, \sigma = 32\%, T = 1) \\ &= \$101e^{-4\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% + \frac{32\%^2}{2}(1 - \frac{1}{25}))}{32\%\sqrt{1 - \frac{1}{25}}}\right) - \$105e^{-2.1\%}\Phi\left(\frac{\ln\frac{\$101}{\$105} + (2.1\% - 4\% - \frac{32\%^2}{2}(1 - \frac{1}{25}))}{32\%\sqrt{1 - \frac{1}{25}}}\right) \\ &= \$9.9987 \end{aligned}$$

The P&L attribution is

$$\delta_5^{reverse} = C_S - C_T = \$9.9987 - \$9.5323 = \$0.4664$$

Python Code and Output:

```python
from scipy.stats import norm
import numpy as np

# Define the Black-Scholes call option price function
def black_scholes_call(S, K, T, r, sigma, q):
    d1 = (np.log(S / K) + (r - q + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    call_price = S * np.exp(-q * T) * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    return call_price

# Initial parameters
K = 105              # Strike price
initial_S = 100      # Initial stock price
final_S = 101        # Final stock price
q = 0.04             # Dividend yield
initial_r = 0.02     # Initial risk-free rate
final_r = 0.021      # Final risk-free rate
initial_sigma = 0.3  # Initial volatility
final_sigma = 0.32   # Final volatility
T = 1                # Time to maturity in years
T_minus_day = T - (1/252)  # Adjust for one day's time decay

# Step 1: Calculate initial option price
initial_price = black_scholes_call(initial_S, K, T, initial_r, initial_sigma, q)

# Perform adjustments in the given order and record the incremental changes
# 1. Adjust stock price
price_after_stock = black_scholes_call(final_S, K, T, initial_r, initial_sigma, q)
delta_stock = price_after_stock - initial_price

# 2. Adjust for time decay (one day less)
price_after_time = black_scholes_call(final_S, K, T_minus_day, initial_r, initial_sigma, q)
delta_time = price_after_time - price_after_stock

# 3. Adjust dividend yield (stays the same, so no change expected)
price_after_dividend = black_scholes_call(final_S, K, T_minus_day, initial_r, initial_sigma, q)
delta_dividend = price_after_dividend - price_after_time

# 4. Adjust risk-free rate
price_after_rate = black_scholes_call(final_S, K, T_minus_day, final_r, initial_sigma, q)
delta_rate = price_after_rate - price_after_dividend

# 5. Adjust volatility
final_price = black_scholes_call(final_S, K, T_minus_day, final_r, final_sigma, q)
delta_volatility = final_price - price_after_rate

# Results in given order
order1_results = {
    'Initial Price': initial_price,
    'After Stock Price Change': price_after_stock,
    'Delta Stock Price': delta_stock,
    'After Time Decay': price_after_time,
    'Delta Time Decay': delta_time,
    'After Dividend Yield': price_after_dividend,
    'Delta Dividend Yield': delta_dividend,
    'After Risk-Free Rate Change': price_after_rate,
    'Delta Risk-Free Rate': delta_rate,
    'After Volatility Change (Final Price)': final_price,
    'Delta Volatility': delta_volatility,
}

# Perform calculations in the reverse order
# 1. Adjust volatility first
price_after_volatility_reverse = black_scholes_call(initial_S, K, T, initial_r, final_sigma, q)
delta_volatility_reverse = price_after_volatility_reverse - initial_price

# 2. Adjust risk-free rate
price_after_rate_reverse = black_scholes_call(initial_S, K, T, final_r, final_sigma, q)
delta_rate_reverse = price_after_rate_reverse - price_after_volatility_reverse

# 3. Adjust dividend yield (stays the same, so no change expected)
price_after_dividend_reverse = black_scholes_call(initial_S, K, T, final_r, final_sigma, q)
delta_dividend_reverse = price_after_dividend_reverse - price_after_rate_reverse

# 4. Adjust for time decay
price_after_time_reverse = black_scholes_call(initial_S, K, T_minus_day, final_r, final_sigma, q)
delta_time_reverse = price_after_time_reverse - price_after_dividend_reverse

# 5. Adjust stock price
final_price_reverse = black_scholes_call(final_S, K, T_minus_day, final_r, final_sigma, q)
delta_stock_reverse = final_price_reverse - price_after_time_reverse

# Results in reverse order
order2_results = {
    'Initial Price': initial_price,
    'After Volatility Change': price_after_volatility_reverse,
    'Delta Volatility': delta_volatility_reverse,
    'After Risk-Free Rate Change': price_after_rate_reverse,
    'Delta Risk-Free Rate': delta_rate_reverse,
    'After Dividend Yield': price_after_dividend_reverse,
    'Delta Dividend Yield': delta_dividend_reverse,
    'After Time Decay': price_after_time_reverse,
    'Delta Time Decay': delta_time_reverse,
    'After Stock Price Change (Final Price)': final_price_reverse,
    'Delta Stock Price': delta_stock_reverse,
}

# Format the results to 4 decimal points and display
order1_results_rounded = {k: round(v, 4) for k, v in order1_results.items()}
order2_results_rounded = {k: round(v, 4) for k, v in order2_results.items()}
order1_results_rounded, order2_results_rounded, round(final_price, 4), round(final_price_reverse, 4)
```

```
({'Initial Price': 8.751,
  'After Stock Price Change': 9.2074,
  'Delta Stock Price': 0.4564,
  'After Time Decay': 9.1888,
  'Delta Time Decay': -0.0186,
  'After Dividend Yield': 9.1888,
  'Delta Dividend Yield': 0.0,
  'After Risk-Free Rate Change': 9.2262,
  'Delta Risk-Free Rate': 0.0374,
  'After Volatility Change (Final Price)': 9.9987,
  'Delta Volatility': 0.7724},
 {'Initial Price': 8.751,
  'After Volatility Change': 9.5159,
  'Delta Volatility': 0.7649,
  'After Risk-Free Rate Change': 9.5523,
  'Delta Risk-Free Rate': 0.0365,
  'After Dividend Yield': 9.5523,
  'Delta Dividend Yield': 0.0,
  'After Time Decay': 9.5323,
  'Delta Time Decay': -0.0201,
  'After Stock Price Change (Final Price)': 9.9987,
  'Delta Stock Price': 0.4664},
 9.9987,
 9.9987)
```

Conclusion:

Looking into more decimal points in both cases, the P&L attribution numbers in the original order and in the reverse order are *different*. When changing the same parameter in the original order and in the reverse order, the other parameters can be different, resulting in different P&L attribution values even though the changing parameter is the same. This discrepancy shows that the sensitivities for a Black-Scholes call also depend on other parameters.

2 Some Analytical Results for ES and VaR

Assuming the losses follow a normal distribution with mean μ and standard deviation σ, derive the formulae for $\text{VaR}_\alpha^{(normal)}$ and $\text{ES}_\alpha^{(normal)}$:

$$\text{VaR}_\alpha^{(normal)} = \mu + \Phi^{-1}(\alpha)\sigma, \tag{1}$$

$$\text{ES}_\alpha^{(normal)} = \mu + \frac{\phi\left(\Phi^{-1}(\alpha)\right)}{1-\alpha}\sigma. \tag{2}$$

Then evaluate the limit:

$$\lim_{\alpha\to 1} \frac{\text{ES}_\alpha^{(normal)}}{\text{VaR}_\alpha^{(normal)}} \tag{3}$$

Now suppose instead that the losses are Pareto distributed. The Pareto distribution has pdf:

$$f(x) = \begin{cases} \frac{\nu\tilde{x}^\nu}{x^{\nu+1}}, & x \geq \tilde{x} \\ 0, & x < \tilde{x} \end{cases} \tag{4}$$

Find expressions for $\text{VaR}_\alpha^{(Pareto)}$ and $\text{ES}_\alpha^{(Pareto)}$. Finally compute the limit

$$\lim_{\alpha\to 1} \frac{\text{ES}_\alpha^{(Pareto)}}{\text{VaR}_\alpha^{(Pareto)}} \tag{5}$$

and compare it to the result for the normal distribution. What does the difference tell you about the tail measures of the two distributions?

Answer:

Normal Distribution:

By definition, VaR_α is the potential loss that a portfolio can suffer for a given confidence level α and time horizon. That is, at a confidence level of α, the losses will not exceed VaR_α. For a loss distribution with fixed time horizon and cumulative distribution function (CDF) F:

$$\text{VaR}_\alpha = \inf\{x \in \mathbb{R} : F(x) \geq \alpha\}$$

Let Φ denote the CDF of a normal distribution, where $\Phi(z)$ is the CDF of a standard normal distribution with mean 0 and standard deviation 1.

Assume that the loss L follow a normal distribution with mean μ and standard deviation σ, i.e., $L \sim N(\mu, \sigma^2)$.

We want to find VaR_α such that

$$\begin{aligned}
&\Pr\left(L \leq \text{VaR}_\alpha\right) = \alpha \quad \text{by definition of VaR} \\
\Rightarrow\ &\Pr\left(\frac{L-\mu}{\sigma} \leq \frac{\text{VaR}_\alpha - \mu}{\sigma}\right) = \alpha \quad \text{by subtracting } \mu \text{ and dividing } \sigma \text{ on both sides of the inequality} \\
\Rightarrow\ &\Pr\left(z \leq \frac{\text{VaR}_\alpha - \mu}{\sigma}\right) = \alpha \quad \text{since } \frac{L-\mu}{\sigma} \sim N(0,1) \\
\Rightarrow\ &\Phi\left(\frac{\text{VaR}_\alpha - \mu}{\sigma}\right) = \alpha \quad \text{by definition of CDF} \\
\Rightarrow\ &\frac{\text{VaR}_\alpha - \mu}{\sigma} = \Phi^{-1}(\alpha) \quad \text{by inverse of CDF} \\
\Rightarrow\ &\text{VaR}_\alpha^{(normal)} = \mu + \Phi^{-1}(\alpha)\sigma \quad \text{arriving at equation (1)} \quad \square
\end{aligned}$$

ES_α is the expected value of loss conditional on losses exceeding VaR_α. We calculate

$$\begin{aligned}
\mathrm{ES}_\alpha^{(normal)} &= \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_x^{(normal)} dx \quad \text{by definition of } \mathrm{ES}_\alpha \\
&= \frac{1}{1-\alpha}\int_\alpha^1 \left(\mu + \Phi^{-1}(x)\sigma\right) dx \quad \text{from equation (1)} \\
&= \frac{1}{1-\alpha}\left(\int_\alpha^1 \mu dx + \int_\alpha^1 \Phi^{-1}(x)\sigma dx\right) \quad \text{by linearity of integral} \\
&= \frac{1}{1-\alpha}\left(\mu(1-\alpha) + \int_\alpha^1 \Phi^{-1}(x)\sigma dx\right) \quad \text{since } \mu \text{ is constant} \\
&= \mu + \frac{\sigma}{1-\alpha}\int_\alpha^1 \Phi^{-1}(x) dx \quad \text{since } \sigma \text{ is constant}
\end{aligned}$$

By change of variable, we set $l = \Phi^{-1}(x) \Rightarrow x = \Phi(x)$.

Taking derivative, we have $\frac{dx}{dl} = \frac{d}{dl}\Phi(l) = \phi(l) \Rightarrow dx = \phi(l)dl$. Then we have

$$\mathrm{ES}_\alpha^{(normal)} = \mu + \frac{\sigma}{1-\alpha}\int_{\Phi^{-1}(\alpha)}^\infty l\phi(l)dl$$

Since $\phi(l)$ represents the probability density function (p.d.f) of the standard normal distribution, given by

$$\phi(l) = \frac{1}{\sqrt{2\pi}}e^{-\frac{l^2}{2}},$$

whose derivative is

$$\phi'(l) = -\frac{2l}{2}\frac{1}{\sqrt{2\pi}}e^{-\frac{l^2}{2}} = -l\phi(l) \Rightarrow \int l\phi(l)dl = -\phi(l)$$

Then we have

$$\begin{aligned}
\mathrm{ES}_\alpha^{(normal)} &= \mu + \frac{\sigma}{1-\alpha}\left[-\phi(l)\right]_{\Phi^{-1}(\alpha)}^\infty \\
&= \mu + \frac{\sigma}{1-\alpha}\left\{0 - \left[-\phi\left(\Phi^{-1}(\alpha)\right)\right]\right\} \\
&= \mu + \frac{\phi\left(\Phi^{-1}(\alpha)\right)}{1-\alpha}\sigma \quad \text{arriving at equation (2)} \quad \square
\end{aligned}$$

Then we evaluate the limit:

$$\begin{aligned}
\lim_{\alpha\to 1}\frac{\mathrm{ES}_\alpha^{(normal)}}{\mathrm{VaR}_\alpha^{(normal)}} &= \lim_{\alpha\to 1}\frac{\mu + \frac{\phi\left(\Phi^{-1}(\alpha)\right)}{1-\alpha}\sigma}{\mu + \Phi^{-1}(\alpha)\sigma} \\
&= \lim_{z\to+\infty}\frac{\mu + \frac{\phi(z)}{1-\Phi(z)}\sigma}{\mu + z\sigma} \quad \text{since } z = \Phi^{-1}(\alpha) \to +\infty \text{ as } \alpha \to 1
\end{aligned}$$

Since $\phi(z) \to 0$ and $1 - \Phi(z) \to 0$ as $z \to \infty$, we can apply L'Hôpital's Rule and compute

$$\begin{aligned}
\lim_{z\to+\infty}\frac{\phi(z)}{1-\Phi(z)} &= \lim_{z\to+\infty}\frac{-z\phi(z)}{-\phi(z)} \\
&= \lim_{z\to+\infty} z \quad (*)
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\lim_{\alpha\to 1}\frac{\mathrm{ES}_\alpha^{(normal)}}{\mathrm{VaR}_\alpha^{(normal)}} &= \lim_{z\to+\infty}\frac{\mu + \frac{\phi(z)}{1-\Phi(z)}\sigma}{\mu + z\sigma} \\
&= \lim_{z\to+\infty}\frac{\mu + z\sigma}{\mu + z\sigma} \quad \text{by substituting } (*) \\
&= 1
\end{aligned}$$

Pareto Distribution:

For $x \le \tilde{x}$, the CDF is $F(x) = 0$.

For $x \ge \tilde{x}$, the CDF is $F(x) = \int_{\tilde{x}}^{x} \frac{v\tilde{x}^v}{t^{v+1}}dt = v\tilde{x}^v \int_{\tilde{x}}^{x} t^{-v-1}dt = v\tilde{x}^v \left[-\frac{1}{v}t^{-v}\right]_{\tilde{x}}^{x} = v\tilde{x}^v \left(-\frac{1}{vx^v} + \frac{1}{v\tilde{x}^v}\right) = 1 - \left(\frac{\tilde{x}}{x}\right)^v$.

Thus, the CDF of the Pareto distribution is

$$F(x) = \Pr(X \le x) = \begin{cases} 1 - \left(\frac{\tilde{x}}{x}\right)^v, & x \ge \tilde{x} \\ 0, & x \le \tilde{x} \end{cases}$$

VaR at confidence α for the Pareto distribution is a quantile that satisfies

$$\begin{aligned} &\Pr(X \le \mathrm{VaR}_\alpha) = \alpha \\ \Rightarrow\ & 1 - \left(\frac{\tilde{x}}{\mathrm{VaR}_\alpha}\right)^v = \alpha \\ \Rightarrow\ & \left(\frac{\tilde{x}}{\mathrm{VaR}_\alpha}\right)^v = 1 - \alpha \\ \Rightarrow\ & \frac{\tilde{x}}{\mathrm{VaR}_\alpha} = (1-\alpha)^{\frac{1}{v}} \\ \Rightarrow\ & \mathrm{VaR}_\alpha^{(Pareto)} = \tilde{x}(1-\alpha)^{-\frac{1}{v}} \end{aligned}$$

The expected loss given that the loss exceeds the VaR at confidence level α for the Pareto distribution is

$$\begin{aligned} \mathrm{ES}_\alpha^{(Pareto)} &= \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_x^{(Pareto)} dx \\ &= \frac{1}{1-\alpha}\int_\alpha^1 \tilde{x}(1-\alpha)^{-\frac{1}{v}} dx \\ &= \frac{\tilde{x}}{1-\alpha}\int_{1-\alpha}^0 u^{-\frac{1}{v}}(-du) \quad \text{letting } u = 1-x \text{, thus } dx = -du \Rightarrow du = -dx \\ &= \frac{\tilde{x}}{1-\alpha}\int_0^{1-\alpha} u^{-\frac{1}{v}} du \\ &= \frac{\tilde{x}}{1-\alpha}\left[\frac{1}{-\frac{1}{v}+1}u^{-\frac{1}{v}+1}\right]_0^{1-\alpha} \\ &= \frac{\tilde{x}}{1-\alpha}\left(\frac{(1-\alpha)^{-\frac{1}{v}+1}}{\frac{v-1}{v}}\right) \\ &= \frac{v}{v-1}\tilde{x}(1-\alpha)^{-\frac{1}{v}} \end{aligned}$$

Finally, we compute the limit for the Pareto distribution:

$$\lim_{\alpha \to 1} \frac{\mathrm{ES}_\alpha^{(Pareto)}}{\mathrm{VaR}_\alpha^{(Pareto)}} = \lim_{\alpha \to 1} \frac{\frac{v}{v-1}\tilde{x}(1-\alpha)^{-\frac{1}{v}}}{\tilde{x}(1-\alpha)^{-\frac{1}{v}}} = \frac{v}{v-1} > 1 = \lim_{\alpha \to 1} \frac{\mathrm{ES}_\alpha^{(normal)}}{\mathrm{VaR}_\alpha^{(normal)}}$$

Comparison:

1. $\lim_{\alpha \to 1} \frac{\mathrm{ES}_\alpha^{(Pareto)}}{\mathrm{VaR}_\alpha^{(Pareto)}} > \lim_{\alpha \to 1} \frac{\mathrm{ES}_\alpha^{(normal)}}{\mathrm{VaR}_\alpha^{(normal)}}$:

   Pareto ES to VaR ratio converges to a *larger* value than normal ES to VaR ratio, indicating the tail measure of Pareto distribution is heavier than that of normal distribution, that is, the Pareto distributed extreme losses are more probable, compared to that the normal distributed extreme losses are rarer.

2. $\lim_{\alpha\to 1} \frac{\mathrm{ES}_\alpha^{(Pareto)}}{\mathrm{VaR}_\alpha^{(Pareto)}} > 1$ and $\lim_{\alpha\to 1} \frac{\mathrm{ES}_\alpha^{(normal)}}{\mathrm{VaR}_\alpha^{(normal)}} = 1$:

   Pareto ES is *higher* than VaR when the confidence level approaches to 1, while the normal ES is approximately *equal* to VaR when the confidence level approaches to 1. This means that ES for Pareto distributions (heavier-tailed distributions) is a more sensitive tail risk measure than VaR, but ES for normal distributions (thinner-tailed distributions) is a tail risk measure approximately as sensitive as VaR, when it comes to a large enough confidence level.

3 Hedging an Equity Portfolio

Suppose there are two stocks, A and B, as well as an equity index I. The returns of A and B are given by

$$r_A = \beta_A R_I + \epsilon_A \tag{6}$$
$$r_B = \beta_B R_I + \epsilon_B \tag{7}$$

where R_I are the returns of the index, $\epsilon_A \sim N(0, \sigma_A)$ and $\epsilon_B \sim N(0, \sigma_B)$. Furthermore assume R_I, ϵ_A, and ϵ_B are independent. For this problem we will use the following parameters: $\sigma(R_I) = 0.2$, $\sigma_A = 0.15$, $\sigma_B = 0.3$, $\beta_A = 0.5$, $\beta_B = 1.5$. The quoted volatilities are annualized. Finally, assume that the index has no expected return and compute the VaR with a 99% confidence level.

Compute the volatility of A and B, the correlation between A and B, and the correlation between the stocks and the index.

The portfolio is made from 5 units of A, 10 units of B and 5 units of the index. A is valued at \$100, B is valued at \$40, and the index is valued at \$50. Use the Gaussian approximation for VaR to obtain an estimate for the 10-day VaR.

Suppose that the desk wants to buy put options on the index to hedge the downside risk of the portfolio. They purchase European put options with a strike of \$45 and $\Delta = -0.25$. Use the Delta approximation for the P&L of the options and determine how many options should be purchased to minimize the VaR. What is the minimum 10-day VaR in this case?

Answer:

The volatility of r_A is

$$\begin{aligned}
\text{Var}(r_A) &= \text{Var}(\beta_A R_I + \epsilon_A) \\
&= \beta_A^2 \text{Var}(R_I) + \text{Var}(\epsilon_A) \quad \text{since } R_I \text{ and } \epsilon_A \text{ are independent} \\
&= 0.5^2 \cdot 0.2^2 + 0.15^2 \\
&= 0.0325 \\
\Rightarrow \sigma_{r_A} &= \sqrt{0.0325} = 0.1803
\end{aligned}$$

The volatility of r_B is

$$\begin{aligned}
\text{Var}(r_B) &= \text{Var}(\beta_B R_I + \epsilon_B) \\
&= \beta_B^2 \text{Var}(R_I) + \text{Var}(\epsilon_B) \quad \text{since } R_I \text{ and } \epsilon_B \text{ are independent} \\
&= 1.5^2 \cdot 0.2^2 + 0.3^2 \\
&= 0.18 \\
\Rightarrow \sigma_{r_B} &= \sqrt{0.18} = 0.4243
\end{aligned}$$

The correlation between r_A and r_B is

$$\begin{aligned}
\rho_{r_A, r_B} &= \frac{\text{Cov}(r_A, r_B)}{\sigma_{r_A} \sigma_{r_B}} \\
&= \frac{\text{Cov}(\beta_A R_I + \epsilon_A, \beta_B R_I + \epsilon_B)}{\sigma_{r_A} \sigma_{r_B}} \\
&= \frac{\beta_A \beta_B \text{Var}(R_I)}{\sigma_{r_A} \sigma_{r_B}} \quad \text{since } R_I, \epsilon_A, \text{ and } \epsilon_B \text{ are independent} \\
&= \frac{0.5 \cdot 1.5 \cdot 0.2^2}{0.1803 \cdot 0.4243} \\
&= 0.3922
\end{aligned}$$

The correlation between r_A and R_I is

$$\begin{aligned}\rho_{r_A,R_I} &= \frac{\text{Cov}(r_A, R_I)}{\sigma_{r_A}\sigma_{R_I}}\\ &= \frac{\text{Cov}(\beta_A R_I + \epsilon_A, R_I)}{\sigma_{r_A}\sigma_{R_I}}\\ &= \frac{\beta_A \text{Var}(R_I)}{\sigma_{r_A}\sigma_{R_I}} \quad \text{since } R_I \text{ and } \epsilon_A \text{ are independent}\\ &= \frac{0.5 \cdot 0.2^2}{0.1803 \cdot 0.2}\\ &= 0.5547\end{aligned}$$

The correlation between r_B and R_I is

$$\begin{aligned}\rho_{r_B,R_I} &= \frac{\text{Cov}(r_B, R_I)}{\sigma_{r_B}\sigma_{R_I}}\\ &= \frac{\text{Cov}(\beta_B R_I + \epsilon_B, R_I)}{\sigma_{r_B}\sigma_{R_I}}\\ &= \frac{\beta_B \text{Var}(R_I)}{\sigma_{r_B}\sigma_{R_I}} \quad \text{since } R_I \text{ and } \epsilon_B \text{ are independent}\\ &= \frac{1.5 \cdot 0.2^2}{0.4243 \cdot 0.2}\\ &= 0.7071\end{aligned}$$

The values of each asset in the portfolio are

$$\begin{aligned}\text{Value of Stock A: } 5 \times \$100 &= \$500\\ \text{Value of Stock B: } 10 \times \$40 &= \$400\\ \text{Value of Index I: } 5 \times \$50 &= \$250\end{aligned}$$

The total portfolio is $V = \$500 + \$400 + \$250 = \$1150$.

The weights of each asset in the portfolio are

$$\begin{aligned}\text{Weight of Stock A: } w_A &= \frac{\$500}{\$1150} = 0.4348\\ \text{Weight of Stock B: } w_B &= \frac{\$400}{\$1150} = 0.3478\\ \text{Weight of Index I: } w_I &= \frac{\$250}{\$1150} = 0.2174\end{aligned}$$

The annualized portfolio volatility is

$$\begin{aligned}\sigma_P^2 &= w_A^2\sigma_{r_A}^2 + w_B^2\sigma_{r_B}^2 + w_I^2\sigma_{R_I}^2 + 2w_Aw_B\sigma_{r_A}\sigma_{r_B}\rho_{r_A,r_B} + 2w_Aw_I\sigma_{r_A}\sigma_{R_I}\rho_{r_A,R_I} + 2w_Bw_I\sigma_{r_B}\sigma_{R_I}\rho_{r_B,R_I}\\ &= 0.4348^2 \cdot 0.1803^2 + 0.3478^2 \cdot 0.4243^2 + 0.2174^2 \cdot 0.2^2 + 2 \cdot 0.4348 \cdot 0.3478 \cdot 0.1803 \cdot 0.4243 \cdot 0.3922\\ &\quad + 2 \cdot 0.4378 \cdot 0.2174 \cdot 0.1803 \cdot 0.2 \cdot 0.5547 + 2 \cdot 0.3478 \cdot 0.2174 \cdot 0.4243 \cdot 0.2 \cdot 0.7071\\ &= 0.0517\\ \Rightarrow \sigma_P &= 0.2275\end{aligned}$$

Using the Gaussian approximation, the 10-Day VaR is

$$\begin{aligned}\text{VaR}_{0.99} &= V \times z_{0.99} \times \sigma_{\text{10-day}} \\ &= \$1150 \times 2.3263 \times 0.2275 \times \sqrt{\frac{10}{252}} \\ &= \$121.22\end{aligned}$$

The portfolio return for every \$1 change in the index return will change by

$$\Delta_P = w_A\beta_A + w_B\beta_B + w_I = 0.4348 \cdot 0.5 + 0.3478 \cdot 1.5 + 0.2174 = 0.9565$$

To hedge the portfolio exposure, we need to purchase the following number of options to minimize the VaR:

$$\begin{aligned}& N_f \times \Delta_f \times S_I + \Delta_P \times V = 0 \\ \Rightarrow\ & N_f \times \Delta_f \times S_I = -\Delta_P \times V \\ \Rightarrow\ & N_f = -\frac{\Delta_P \times V}{\Delta_f \times S_I} \\ & \quad = -\frac{0.9565 \times \$1150}{-0.25 \times \$50} \\ & \quad = 88\end{aligned}$$

Method 1: Assume residual systematic risk in r_A and r_B:

After hedging the risk of index, the updated annualized portfolio volatility becomes

$$\begin{aligned}\sigma^2_{P,\text{new}} &= w_A^2\sigma^2_{r_A} + w_B^2\sigma^2_{r_B} + 2w_Aw_B\sigma_{r_A}\sigma_{r_B}\rho_{r_A,r_B} \\ &= 0.4348^2 \cdot 0.1803^2 + 0.3478^2 \cdot 0.4243^2 + 2 \cdot 0.4348 \cdot 0.3478 \cdot 0.1803 \cdot 0.4243 \cdot 0.3922 \\ &= 0.037 \\ \Rightarrow \sigma_{P,\text{new}} &= 0.1923\end{aligned}$$

Using Delta approximation, the minimum 10-day VaR in this case is

$$\begin{aligned}\text{VaR}^{\min}_{0.99} &= V \times z_{0.99} \times \sigma_{\text{10-day, new}} \\ &= \$1150 \times 2.3263 \times 0.1923 \times \sqrt{\frac{10}{252}} \\ &= \$102.5\end{aligned}$$

Method 2: Assume fully removed systematic risk in r_A and r_B:

After hedging the risk of index, the updated annualized portfolio volatility becomes

$$\begin{aligned}\sigma^2_{P,\text{new}} &= w_A^2\text{Var}(\epsilon_{r_A}) + w_B^2\text{Var}(\epsilon_{r_B}) \\ &= 0.4348^2 \cdot 0.15^2 + 0.3478^2 \cdot 0.3^2 \\ &= 0.0151 \\ \Rightarrow \sigma_{P,\text{new}} &= 0.1230\end{aligned}$$

Using Delta approximation, the minimum 10-day VaR in this case is

$$\begin{aligned}\text{VaR}^{\min}_{0.99} &= V \times z_{0.99} \times \sigma_{\text{10-day, new}} \\ &= \$1150 \times 2.3263 \times 0.1230 \times \sqrt{\frac{10}{252}} \\ &= \$65.57\end{aligned}$$

Python Code and Output:

```python
import math

# Given parameters
beta_A = 0.5
beta_B = 1.5
sigma_RI = 0.2
sigma_A_epsilon = 0.15
sigma_B_epsilon = 0.3

# Calculate variances and volatilities (standard deviations)
Var_r_A = beta_A**2 * sigma_RI**2 + sigma_A_epsilon**2
sigma_r_A = math.sqrt(Var_r_A)

Var_r_B = beta_B**2 * sigma_RI**2 + sigma_B_epsilon**2
sigma_r_B = math.sqrt(Var_r_B)

# Correlations
# Correlation between r_A and r_B
rho_rA_rB = (beta_A * beta_B * sigma_RI**2) / (sigma_r_A * sigma_r_B)

# Correlation between r_A and R_I
rho_rA_RI = (beta_A * sigma_RI**2) / (sigma_r_A * sigma_RI)

# Correlation between r_B and R_I
rho_rB_RI = (beta_B * sigma_RI**2) / (sigma_r_B * sigma_RI)

# Results with four decimal points
results = {
    "Var(r_A)": round(Var_r_A, 4),
    "sigma_r_A": round(sigma_r_A, 4),
    "Var(r_B)": round(Var_r_B, 4),
    "sigma_r_B": round(sigma_r_B, 4),
    "rho(r_A, r_B)": round(rho_rA_rB, 4),
    "rho(r_A, R_I)": round(rho_rA_RI, 4),
    "rho(r_B, R_I)": round(rho_rB_RI, 4)
}

results
```

```
{'Var(r_A)': 0.0325,
 'sigma_r_A': 0.1803,
 'Var(r_B)': 0.18,
 'sigma_r_B': 0.4243,
 'rho(r_A, r_B)': 0.3922,
 'rho(r_A, R_I)': 0.5547,
 'rho(r_B, R_I)': 0.7071}
```

```python
# Portfolio weights based on asset values
w_A = 500 / 1150  # Value of A: 5 * 100 = 500
w_B = 400 / 1150  # Value of B: 10 * 40 = 400
w_I = 250 / 1150  # Value of Index I: 5 * 50 = 250

# Volatilities (standard deviations) of individual stocks
sigma_r_A = math.sqrt(beta_A**2 * sigma_RI**2 + sigma_A_epsilon**2)
sigma_r_B = math.sqrt(beta_B**2 * sigma_RI**2 + sigma_B_epsilon**2)

# Correlations between assets
```

```
rho_rA_rB = (beta_A * beta_B * sigma_RI**2) / (sigma_r_A * sigma_r_B)
rho_rA_RI = (beta_A * sigma_RI**2) / (sigma_r_A * sigma_RI)
rho_rB_RI = (beta_B * sigma_RI**2) / (sigma_r_B * sigma_RI)

# Portfolio variance components
# Individual variance terms
term_A = w_A**2 * sigma_r_A**2
term_B = w_B**2 * sigma_r_B**2
term_I = w_I**2 * sigma_RI**2

# Cross terms
cross_AB = 2 * w_A * w_B * sigma_r_A * sigma_r_B * rho_rA_rB
cross_AI = 2 * w_A * w_I * sigma_r_A * sigma_RI * rho_rA_RI
cross_BI = 2 * w_B * w_I * sigma_r_B * sigma_RI * rho_rB_RI

# Portfolio variance and volatility
portfolio_variance = term_A + term_B + term_I + cross_AB + cross_AI + cross_BI
portfolio_volatility = math.sqrt(portfolio_variance)

# Print results rounded to 4 decimal places
print("w_A:", round(w_A, 4))
print("w_B:", round(w_B, 4))
print("w_I:", round(w_I, 4))
print("Var(r_A):", round(sigma_r_A**2, 4))
print("sigma_r_A:", round(sigma_r_A, 4))
print("Var(r_B):", round(sigma_r_B**2, 4))
print("sigma_r_B:", round(sigma_r_B, 4))
print("rho(r_A, r_B):", round(rho_rA_rB, 4))
print("rho(r_A, R_I):", round(rho_rA_RI, 4))
print("rho(r_B, R_I):", round(rho_rB_RI, 4))
print("Portfolio Variance:", round(portfolio_variance, 4))
print("Portfolio Volatility:", round(portfolio_volatility, 4))
```

```
w_A: 0.4348
w_B: 0.3478
w_I: 0.2174
Var(r_A): 0.0325
sigma_r_A: 0.1803
Var(r_B): 0.18
sigma_r_B: 0.4243
rho(r_A, r_B): 0.3922
rho(r_A, R_I): 0.5547
rho(r_B, R_I): 0.7071
Portfolio Variance: 0.0517
Portfolio Volatility: 0.2275
```

```
from scipy.stats import norm
import math

# Define parameters
V = 1150  # Portfolio value
time_horizon_days = 10
days_in_year = 252

# Calculate z_0.99 accurately using the inverse CDF (quantile) function for the normal distribution
z_99 = norm.ppf(0.99)

# Calculate the 10-day adjusted portfolio volatility
```

```
sigma_10day = portfolio_volatility * math.sqrt(time_horizon_days / days_in_year)

# Calculate 10-day VaR at 99% confidence level
VaR_10day_99 = V * z_99 * sigma_10day

# Output VaR with two decimal places
print("99% z-score:", round(z_99, 4))
print("10-day VaR at 99% confidence level:", round(VaR_10day_99, 2))
```

```
99% z-score: 2.3263
10-day VaR at 99% confidence level: 121.22
```

```
Delta_f = -0.25
S_I = 50

Delta_P = w_A*beta_A+w_B*beta_B+w_I
print("Delta_P:", round(Delta_P, 4))

N_f = -Delta_P*V/Delta_f/S_I
print("N_f:", round(N_f, 4))

portfolio_variance_new = term_A + term_B + cross_AB
portfolio_volatility_new = np.sqrt(portfolio_variance_new)
print("portfolio_variance_new:", round(portfolio_variance_new, 4))
print("portfolio_volatility_new:", round(portfolio_volatility_new, 4))

sigma_10day_new = portfolio_volatility_new * math.sqrt(time_horizon_days / days_in_year)

# Calculate 10-day VaR at 99% confidence level
VaR_10day_99_min = V * z_99 * sigma_10day_new

# Output VaR with two decimal places
print("99% z-score:", round(z_99, 4))
print("Minimum 10-day VaR at 99% confidence level:", round(VaR_10day_99_min, 2))
```

```
Delta_P: 0.9565
N_f: 88.0
portfolio_variance_new: 0.037
portfolio_volatility_new: 0.1923
99% z-score: 2.3263
Minimum 10-day VaR at 99% confidence level: 102.5
```

```
import numpy as np

# Given data
w_A, w_B = 0.4348, 0.3478  # weights of stocks A and B
sigma_epsilon_A, sigma_epsilon_B = 0.15, 0.3  # idiosyncratic volatilities for A and B
V = 1150  # total portfolio value
z_99 = 2.3263  # z-score for 99% confidence level
days = 10
annual_trading_days = 252

# Calculate new portfolio variance based only on idiosyncratic risks of A and B
sigma_P_new_squared = (w_A**2 * sigma_epsilon_A**2 + w_B**2 * sigma_epsilon_B**2)
sigma_P_new = np.sqrt(sigma_P_new_squared)

# Convert annualized volatility to 10-day volatility
sigma_10_day_new = sigma_P_new * np.sqrt(days / annual_trading_days)
```

```python
# Calculate 10-day VaR
VaR_min_99 = V * z_99 * sigma_10_day_new

# Display results
print("portfolio_variance_new:", sigma_P_new_squared)
print("portfolio_volatility_new:", sigma_P_new)
print("Minimum 10-day VaR at 99% confidence level:", VaR_min_99)
```

```
portfolio_variance_new: 0.015140484
portfolio_volatility_new: 0.12304667407126452
Minimum 10-day VaR at 99% confidence level: 65.57422551894264
```

4 Netting for Correlated Brownian Motions

The bank enters into two contracts with counterparty A. The market-to-market of the contracts are give by:

$$\mathrm{MtM}_1(t) = \mu_1 + \sigma_1 W_1(t) \tag{8}$$
$$\mathrm{MtM}_2(t) = \mu_2 + \sigma_2 W_2(t) \tag{9}$$

for some μ_1, μ_2, σ_1, σ_2 and where W_1 and W_2 are Wiener processes. Furthermore, assume the two processes are correlated with

$$\mathrm{Corr}[W_1(t), W_2(t)] = \rho \tag{10}$$

Compute the expected exposure of the bank to counterparty A assuming the two contracts are not nettable.

Compute the expected exposure of the bank to counterparty A assuming the two contracts are nettable.

Comment on your results (it may be useful to plot some figures to interpret the results).

Write a program to compute the expected exposure and confirm your analytical results from the above. Further more, use the program to compute a PFE profile using the expected shortfall method with the top 10% of exposures for both netted and non-netted cases. Try a few different values of the correlation to get a feel for how that influences the risk.

Answer:

Method 1: Direct Linear Combination with ρ

Since the two Wiener processes are correlated with $\mathrm{Corr}[W_1(t), W_2(t)] = \rho$, we can express $W_2(t)$ in terms of $W_2(t)$ and an independent Wiener process $Z(t)$ as follows:

$$W_2(t) = \rho W_1(t) + \sqrt{1-\rho^2} Z(t), \quad (*)$$

where $W_1(t)$ and $W_2(t)$ satisfy the desired correlation ρ.

To prove this, we compute the correlation between the two Wiener processes:

$$\begin{aligned}
\mathrm{Corr}(W_1(t), W_2(t)) &= \frac{\mathrm{Cov}(W_1(t), W_2(t))}{\sqrt{\mathrm{Var}(W_1(t)) \cdot \mathrm{Var}(W_2(t))}} \\
&= \frac{\mathrm{Cov}\left(W_1(t), \rho W_1(t) + \sqrt{1-\rho^2} Z(t)\right)}{\sqrt{t \cdot t}} \quad \text{since a Wiener process } W(t) \sim N(0,t) \\
&= \frac{\rho \mathrm{Var}(W_1(t))}{t} \quad \text{since } W_1(t) \text{ and } Z(t) \text{ are independent} \\
&= \frac{\rho \cdot t}{t} \\
&= \rho \quad \square
\end{aligned}$$

The exposure at time t for scenario s for not nettable and nettable contracts are

$$\begin{aligned}
E_s^{\text{non-net}}(t) &= \max\left[\mathrm{MtM}_{1,s}(t), 0\right] + \max\left[\mathrm{MtM}_{2,s}(t), 0\right] \\
&= \max\left[\mu_1 + \sigma_1 W_{1,s}(t), 0\right] + \max\left[\mu_2 + \sigma_2 W_{2,s}(t), 0\right] \quad \text{from equation (8) and (9)} \\
&= \max\left[\mu_1 + \sigma_1 W_{1,s}(t), 0\right] + \max\left[\mu_2 + \sigma_2 \left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right] \quad \text{from equation } (*) \\
E_s^{\text{net}}(t) &= \max\left[\mathrm{MtM}_{1,s}(t) + \mathrm{MtM}_{2,s}(t), 0\right] \\
&= \max\left[\mu_1 + \sigma_1 W_{1,s}(t) + \mu_2 + \sigma_2 W_{2,s}(t), 0\right] \quad \text{from equation (8) and (9)} \\
&= \max\left[\mu_1 + \sigma_1 W_{1,s}(t) + \mu_2 + \sigma_2 \left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right] \quad \text{from equation } (*)
\end{aligned}$$

The expected exposure at time t for not nettable and nettable contracts are

$$\begin{aligned}
EE^{\text{non-net}}(t) &= \mathbb{E}[E^{\text{non-net}}(t)] \\
&= \frac{1}{N_s}\sum_{s}^{N_s} E_s^{\text{non-net}}(t) \\
&= \frac{1}{N_s}\sum_{s}^{N_s} \max\left[\mu_1 + \sigma_1 W_{1,s}(t), 0\right] + \max\left[\mu_2 + \sigma_2\left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right] \\
EE^{\text{net}}(t) &= \mathbb{E}[E^{\text{net}}(t)] \\
&= \frac{1}{N_s}\sum_{s}^{N_s} E_s^{\text{net}}(t) \\
&= \frac{1}{N_s}\sum_{s}^{N_s} \max\left[\mu_1 + \sigma_1 W_{1,s}(t) + \mu_2 + \sigma_2\left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right]
\end{aligned}$$

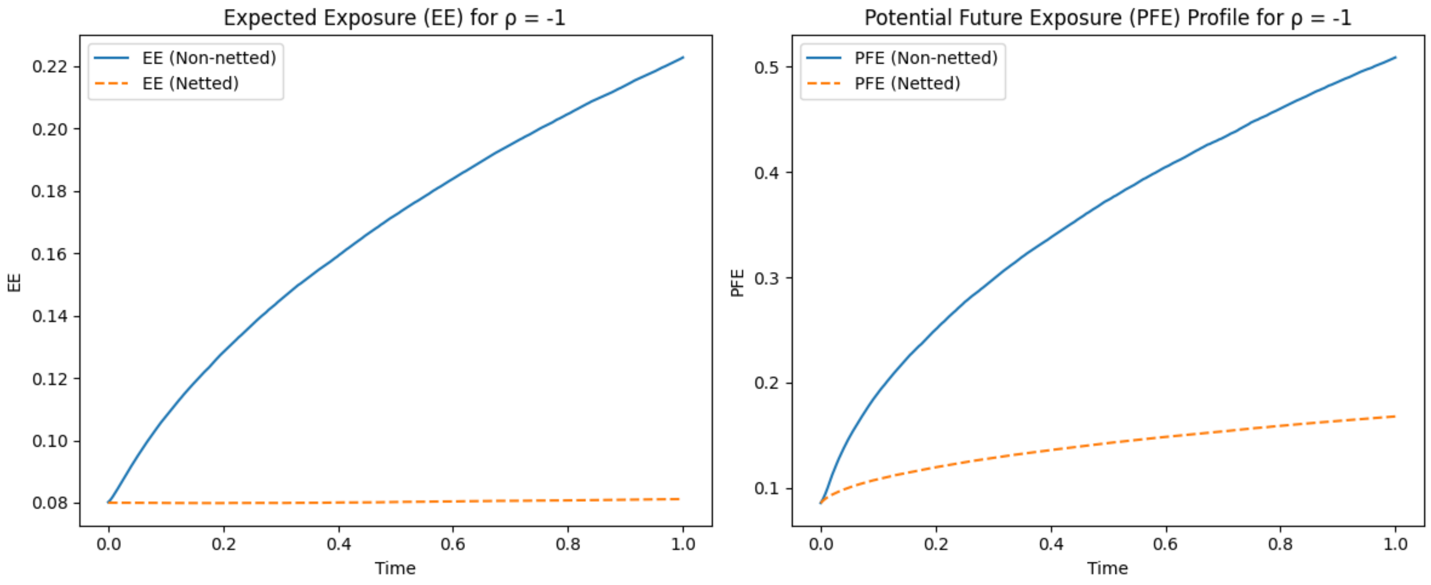
The potential future exposure at time t using the expected shortfall method with the top 10% of exposures for not nettable and nettable contracts are

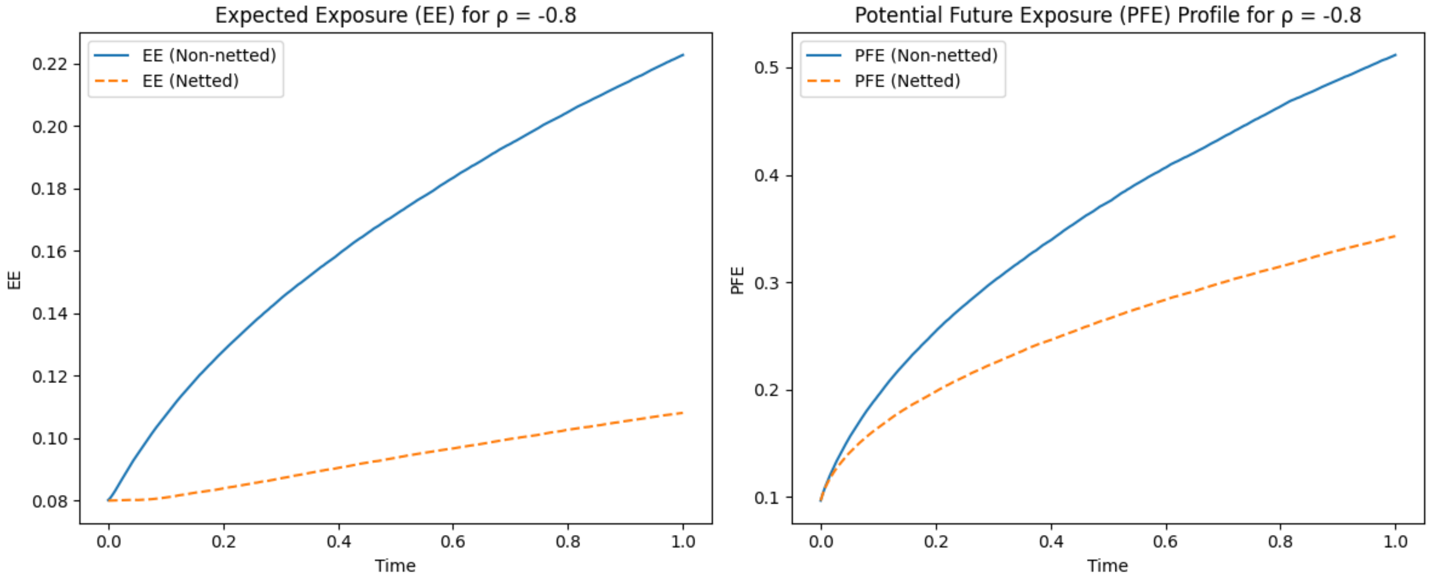
$$\begin{aligned}
PFE^{\text{non-net}}(t) &= \frac{1}{N_{\text{top }10\%}} \sum_{s\in\text{top }10\%} E_s^{\text{non-net}}(t) \\
&= \frac{1}{N_{\text{top }10\%}} \sum_{s\in\text{top }10\%} \max\left[\mu_1 + \sigma_1 W_{1,s}(t), 0\right] + \max\left[\mu_2 + \sigma_2\left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right] \\
PFE^{\text{net}}(t) &= \frac{1}{N_{\text{top }10\%}} \sum_{s\in\text{top }10\%} E_s^{\text{net}}(t) \\
&= \frac{1}{N_{\text{top }10\%}} \sum_{s\in\text{top }10\%} \max\left[\mu_1 + \sigma_1 W_{1,s}(t) + \mu_2 + \sigma_2\left(\rho W_{1,s}(t) + \sqrt{1-\rho^2} Z_s(t)\right), 0\right]
\end{aligned}$$

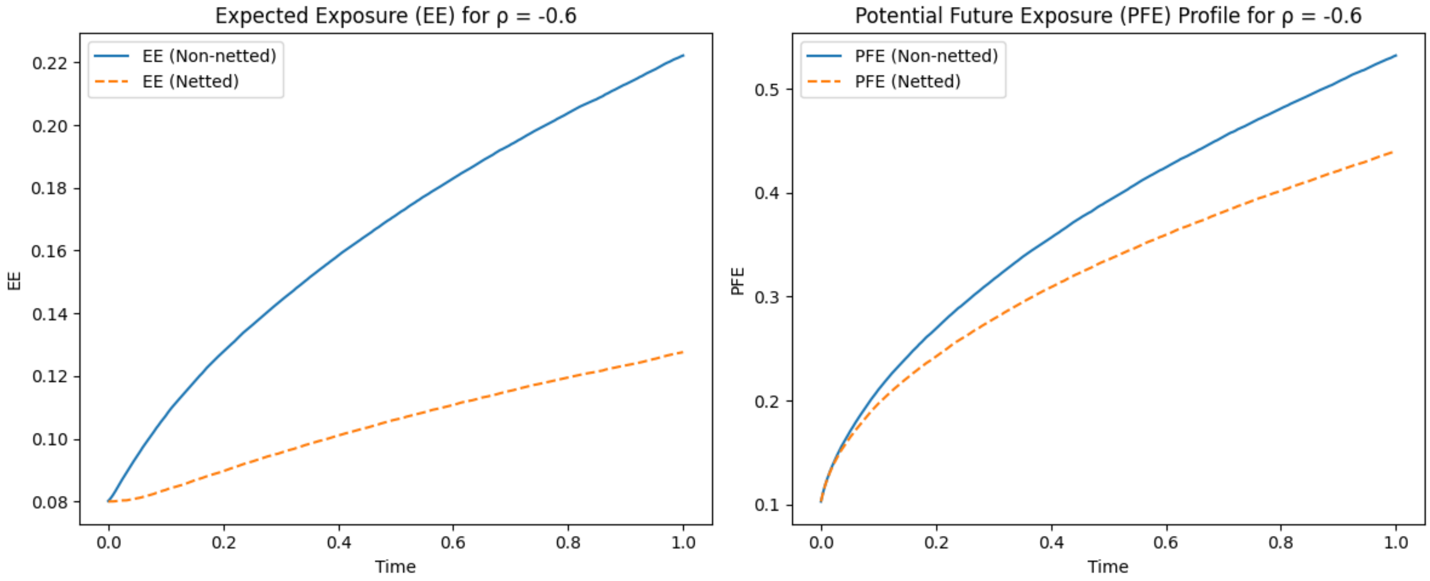
Python Code and Output:

Assuming $\mu_1 = 0.05$, $\mu_2 = 0.03$, $\sigma_1 = 0.2$, $\sigma_2 = 0.25$, $T = 1$, $dt = \frac{1}{252}$, and $N_s = 10000$, we program in Python to plot not nettable and nettable daily expected exposures (EE) as well as not nettable and nettable daily potential future exposure (PFE) profile using the expected shortfall method with the top 10% of exposures for different correlation values, $\rho = -1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1$, over 1-year time horizon.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
np.random.seed(1005778428)

# Parameters
mu1, mu2 = 0.05, 0.03  # expected returns for contracts 1 and 2
sigma1, sigma2 = 0.2, 0.25  # volatilities for contracts 1 and 2
T = 1  # time horizon in years
dt = 1/252  # time step (daily)
n_steps = int(T / dt)
n_simulations = 100000  # number of Monte Carlo simulations
correlations = [-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1]  # different correlation values

# Function to generate correlated Wiener processes
def generate_correlated_wiener_processes(rho, n_simulations, n_steps):
    W1 = np.random.normal(0, np.sqrt(dt), (n_simulations, n_steps))
    W2 = rho * W1 + np.sqrt(1 - rho**2) * np.random.normal(0, np.sqrt(dt), (n_simulations, n_steps))
    return W1.cumsum(axis=1), W2.cumsum(axis=1)

# Function to calculate exposures
def calculate_exposures(rho):
    W1, W2 = generate_correlated_wiener_processes(rho, n_simulations, n_steps)
    MtM1 = mu1 + sigma1 * W1
    MtM2 = mu2 + sigma2 * W2

    # Non-netted exposure
    E_non_net = np.maximum(MtM1, 0) + np.maximum(MtM2, 0)
    EE_non_net = E_non_net.mean(axis=0)
    # Calculate PFE as the mean of the top 10% exposures at each time step
    PFE_non_net = np.array([np.mean(E_non_net[:, i][E_non_net[:, i] >=
                  np.percentile(E_non_net[:, i], 90)]) for i in range(n_steps)])

    # Netted exposure
    E_net = np.maximum(MtM1 + MtM2, 0)
    EE_net = E_net.mean(axis=0)
    # Calculate PFE as the mean of the top 10% exposures at each time step
    PFE_net = np.array([np.mean(E_net[:, i][E_net[:, i] >=
              np.percentile(E_net[:, i], 90)]) for i in range(n_steps)])

    return EE_non_net, PFE_non_net, EE_net, PFE_net

# Plotting function
def plot_exposures(EE_non_net, PFE_non_net, EE_net, PFE_net, rho):
    time_grid = np.linspace(0, T, n_steps)

    plt.figure(figsize=(12, 5))

    # Plot for Expected Exposure (EE)
    plt.subplot(1, 2, 1)
    plt.plot(time_grid, EE_non_net, label='EE (Non-netted)', linestyle='-', linewidth=1.5)
    plt.plot(time_grid, EE_net, label='EE (Netted)', linestyle='--', linewidth=1.5)
    plt.title(f'Expected Exposure (EE) for  = {rho}')
    plt.xlabel('Time')
    plt.ylabel('EE')
    plt.legend()

    # Plot for Potential Future Exposure (PFE)
    plt.subplot(1, 2, 2)
    plt.plot(time_grid, PFE_non_net, label='PFE (Non-netted)', linestyle='-', linewidth=1.5)
    plt.plot(time_grid, PFE_net, label='PFE (Netted)', linestyle='--', linewidth=1.5)
    plt.title(f'Potential Future Exposure (PFE) Profile for  = {rho}')
    plt.xlabel('Time')
    plt.ylabel('PFE')
    plt.legend()

    plt.tight_layout()
    plt.show()

# Run simulations, plot, and capture final values for each correlation
final_values = []

for rho in correlations:
    EE_non_net, PFE_non_net, EE_net, PFE_net = calculate_exposures(rho)
    plot_exposures(EE_non_net, PFE_non_net, EE_net, PFE_net, rho)

    final_EE_non_net = EE_non_net[-1]
    final_PFE_non_net = PFE_non_net[-1]
    final_EE_net = EE_net[-1]
    final_PFE_net = PFE_net[-1]

    final_values.append({
        'Correlation ()': rho,
        'Final EE (Non-netted)': final_EE_non_net,
        'Final PFE (Non-netted)': final_PFE_non_net,
        'Final EE (Netted)': final_EE_net,
        'Final PFE (Netted)': final_PFE_net
    })

# Create dataframe and round values to 4 decimal points
final_values_df = pd.DataFrame(final_values).round(4)

# Display final results
print(final_values_df)
```

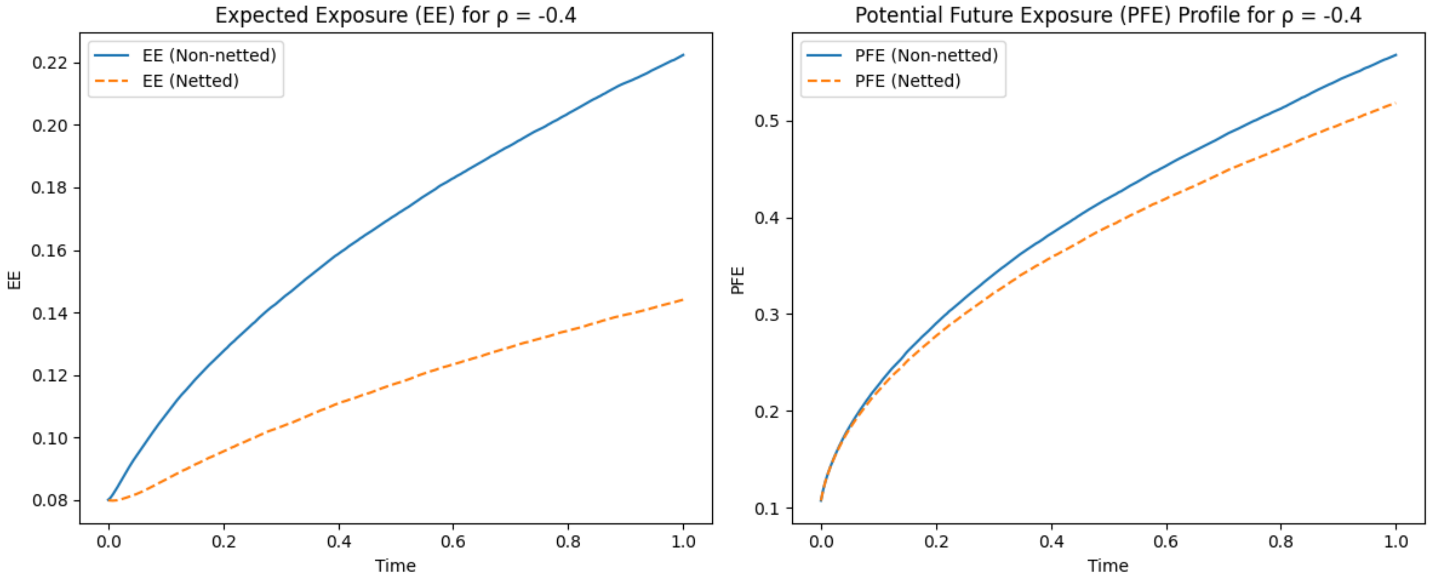






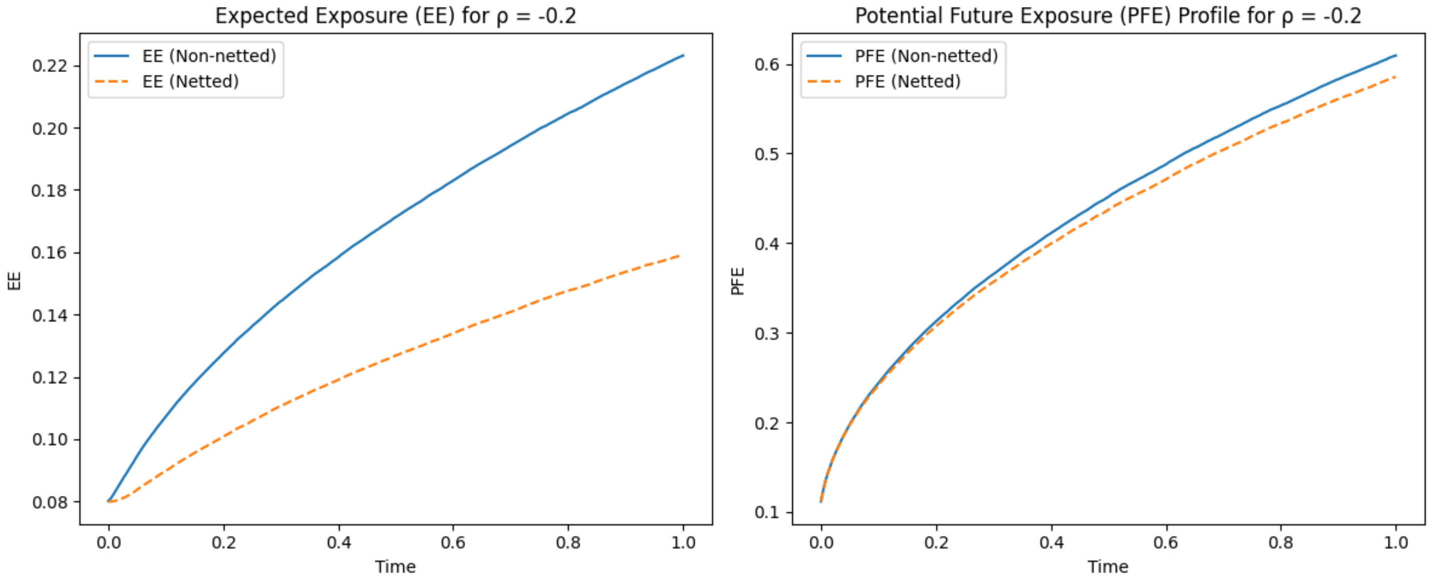
Expected Exposure (EE) for ρ = -0.6

Potential Future Exposure (PFE) Profile for ρ = -0.6



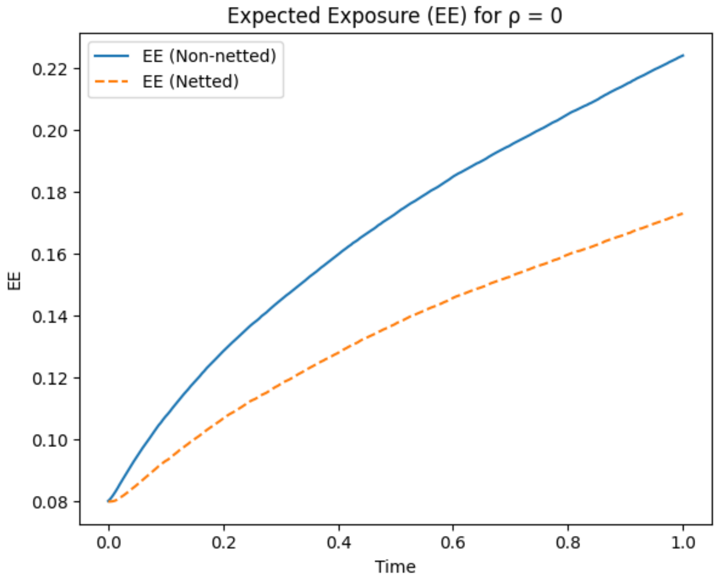
Expected Exposure (EE) for ρ = -0.4

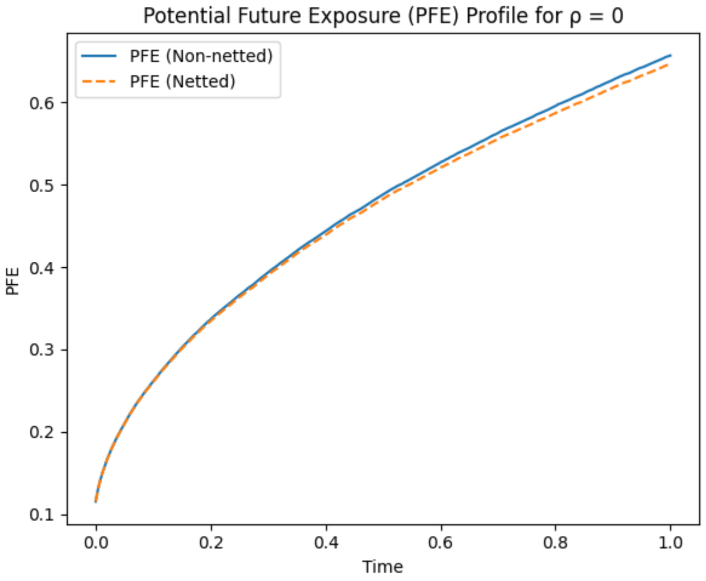
Potential Future Exposure (PFE) Profile for ρ = -0.4

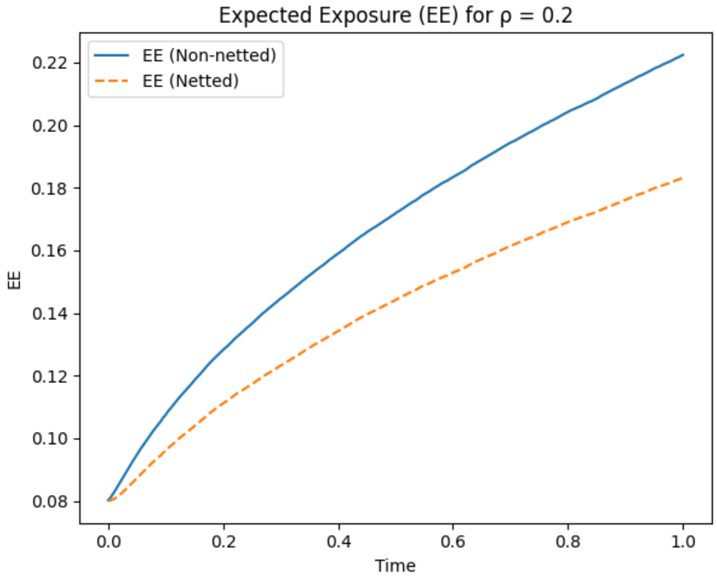


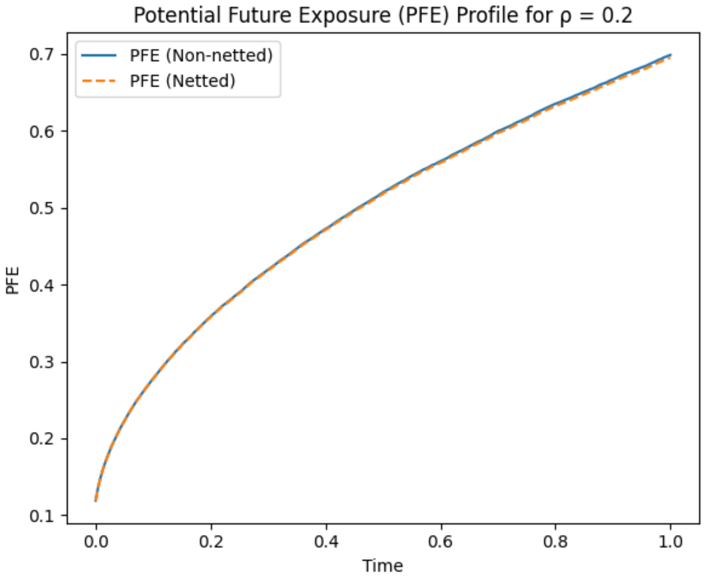
Expected Exposure (EE) for ρ = -0.2

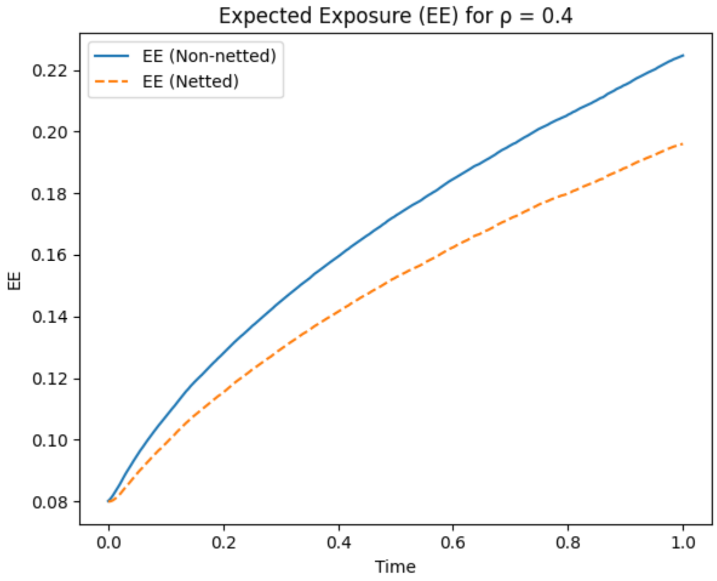
Potential Future Exposure (PFE) Profile for ρ = -0.2

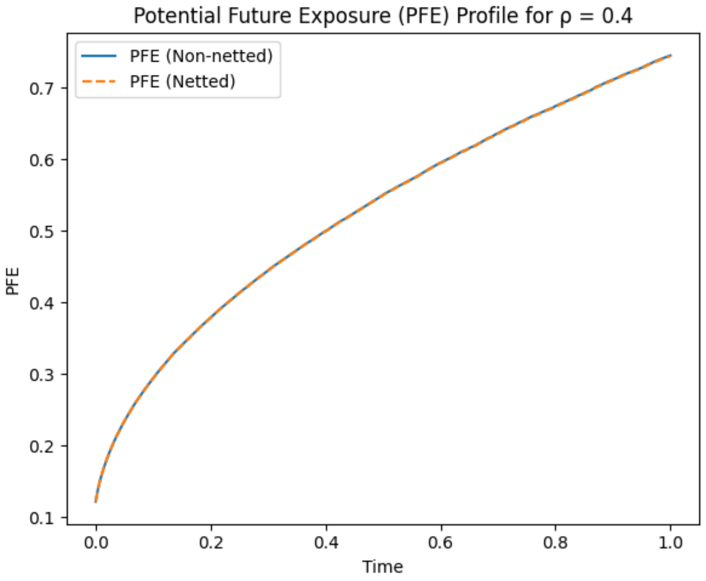

Expected Exposure (EE) for ρ = 0
EE (Non-netted)
EE (Netted)
EE
Time
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0

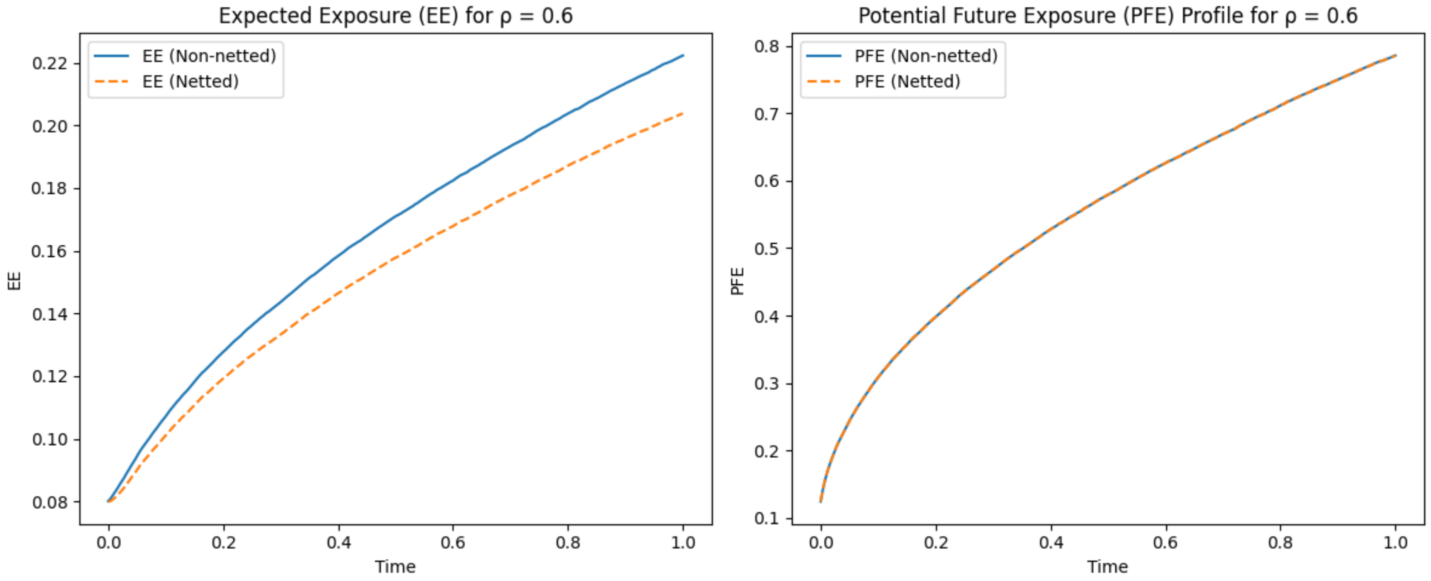

Potential Future Exposure (PFE) Profile for ρ = 0
PFE (Non-netted)
PFE (Netted)
PFE
Time
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0

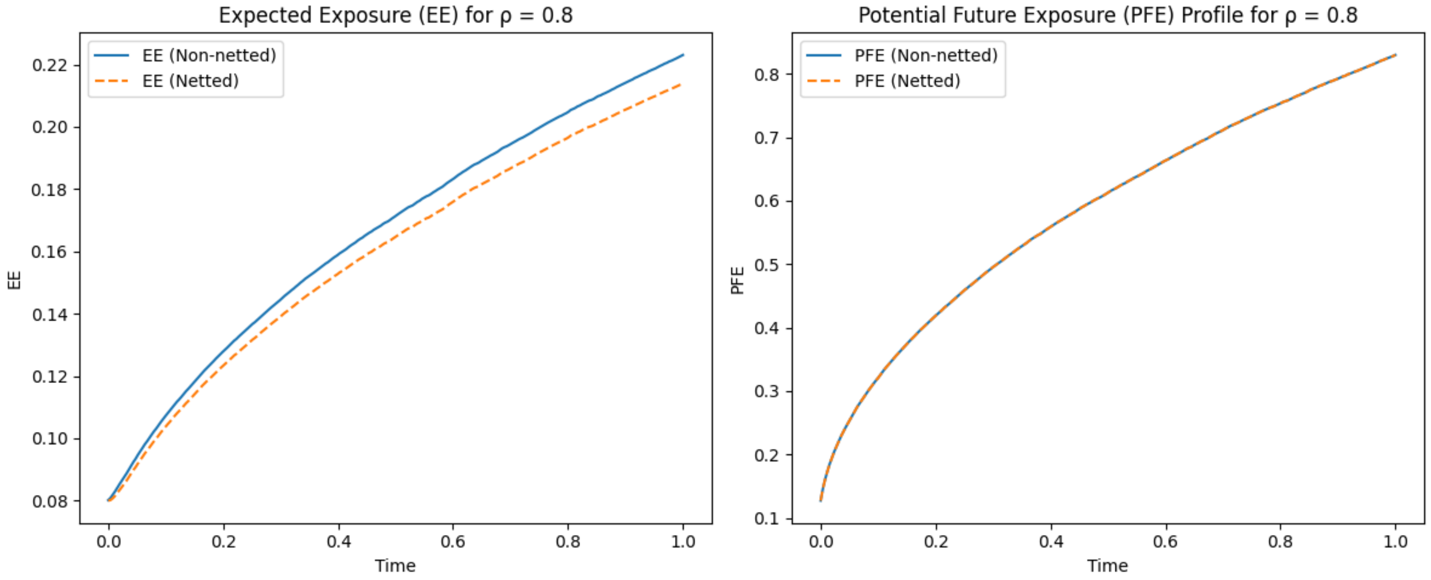

Expected Exposure (EE) for ρ = 0.2
EE (Non-netted)
EE (Netted)
EE
Time
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0

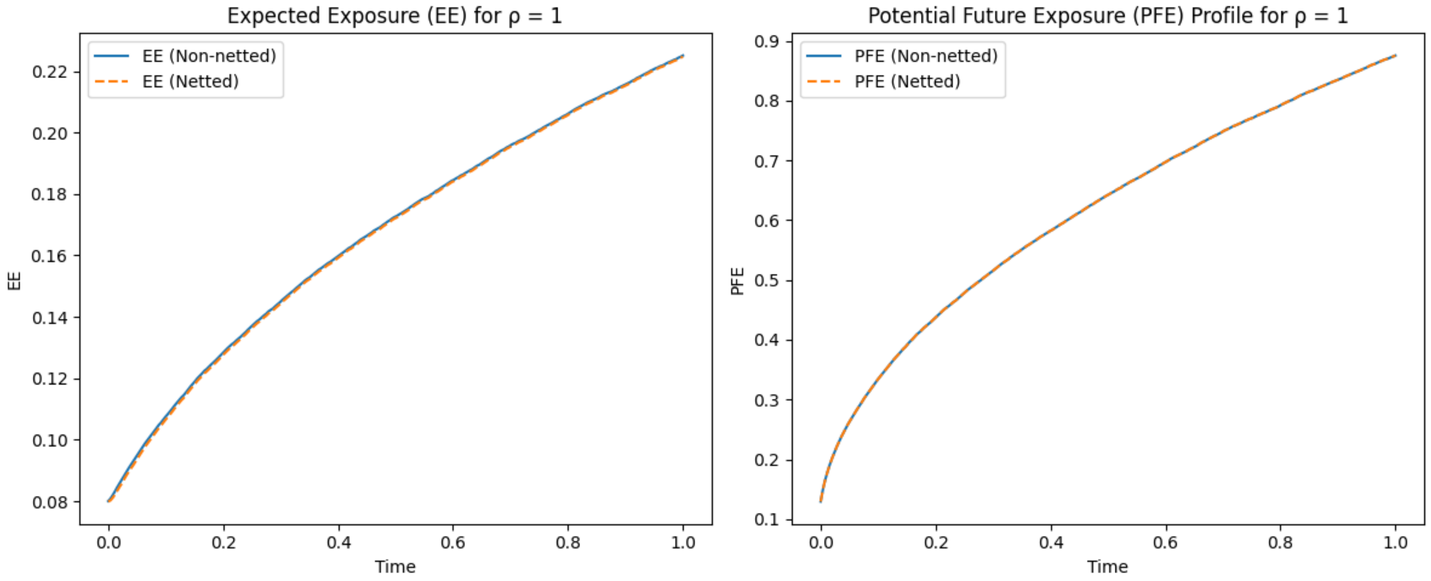

Potential Future Exposure (PFE) Profile for ρ = 0.2
PFE (Non-netted)
PFE (Netted)
PFE
Time
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8
1.0


Expected Exposure (EE) for ρ = 0.4
EE (Non-netted)
EE (Netted)
EE
Time
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0


Potential Future Exposure (PFE) Profile for ρ = 0.4
PFE (Non-netted)
PFE (Netted)
PFE
Time
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8
1.0


Expected Exposure (EE) for ρ = 0.6
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 0.6
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
0.2
0.4
0.6
0.8
1.0
Time


Expected Exposure (EE) for ρ = 0.8
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 0.8
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
0.2
0.4
0.6
0.8
1.0
Time


Expected Exposure (EE) for ρ = 1
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 1
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.0
0.2
0.4
0.6
0.8
1.0
Time

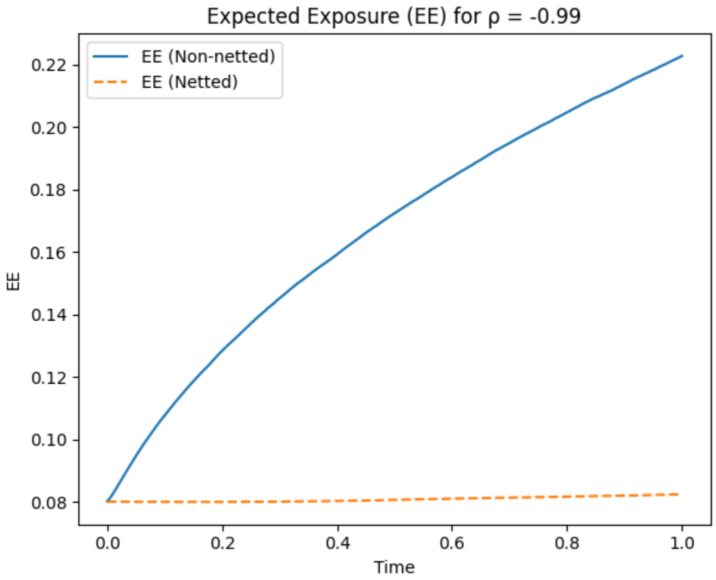
Compute non-netted and netted EE as well as non-netted and netted PFE at time $T = 1$:

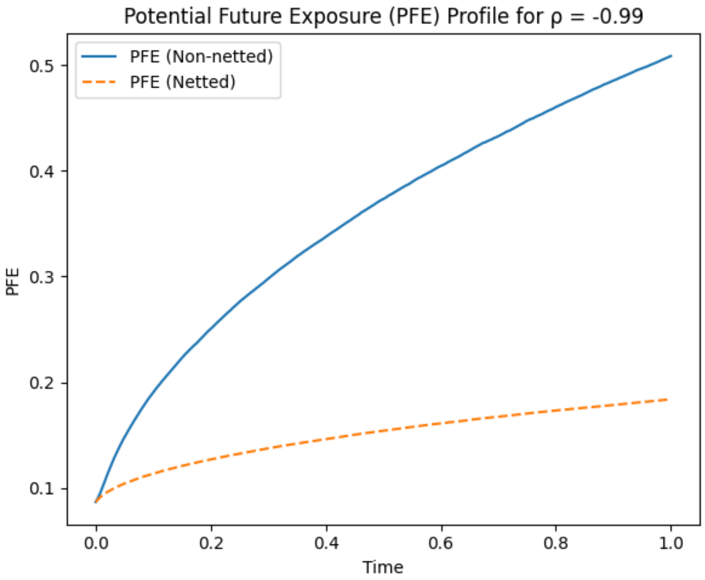
Correlation (ρ)	Final EE (Non-netted)	Final EE (Netted)	Final PFE (Non-netted)	Final PFE (Netted)
-1.0	0.2228	0.0812	0.5088	0.1679
-0.8	0.2228	0.1081	0.5116	0.3429
-0.6	0.2222	0.1276	0.5321	0.4399
-0.4	0.2224	0.1441	0.5675	0.5179
-0.2	0.2232	0.1591	0.6092	0.5853
0.0	0.2240	0.1729	0.6565	0.6461
0.2	0.2224	0.1832	0.6988	0.6953
0.4	0.2247	0.1960	0.7454	0.7448
0.6	0.2223	0.2038	0.7854	0.7853
0.8	0.2231	0.2139	0.8299	0.8299
1.0	0.2250	0.2247	0.8753	0.8753

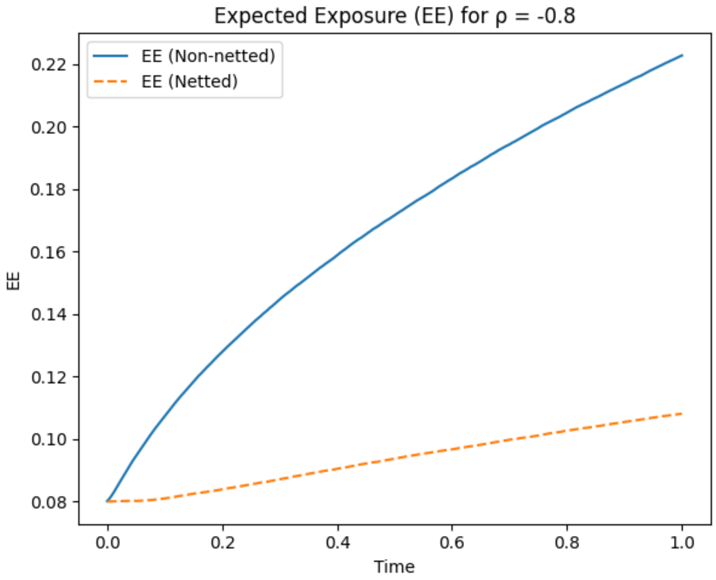
Table 1: Expected Exposure (EE) and Potential Future Exposure (PFE) values for different correlation levels.

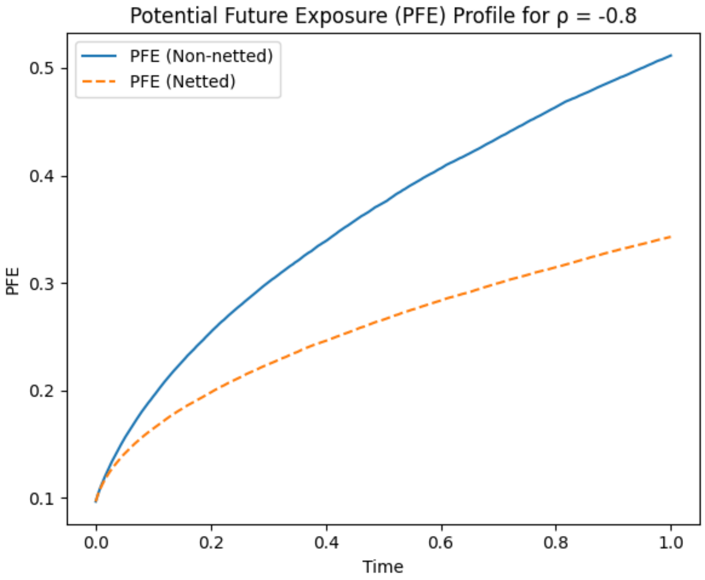
Method 2: Cholesky Decomposition

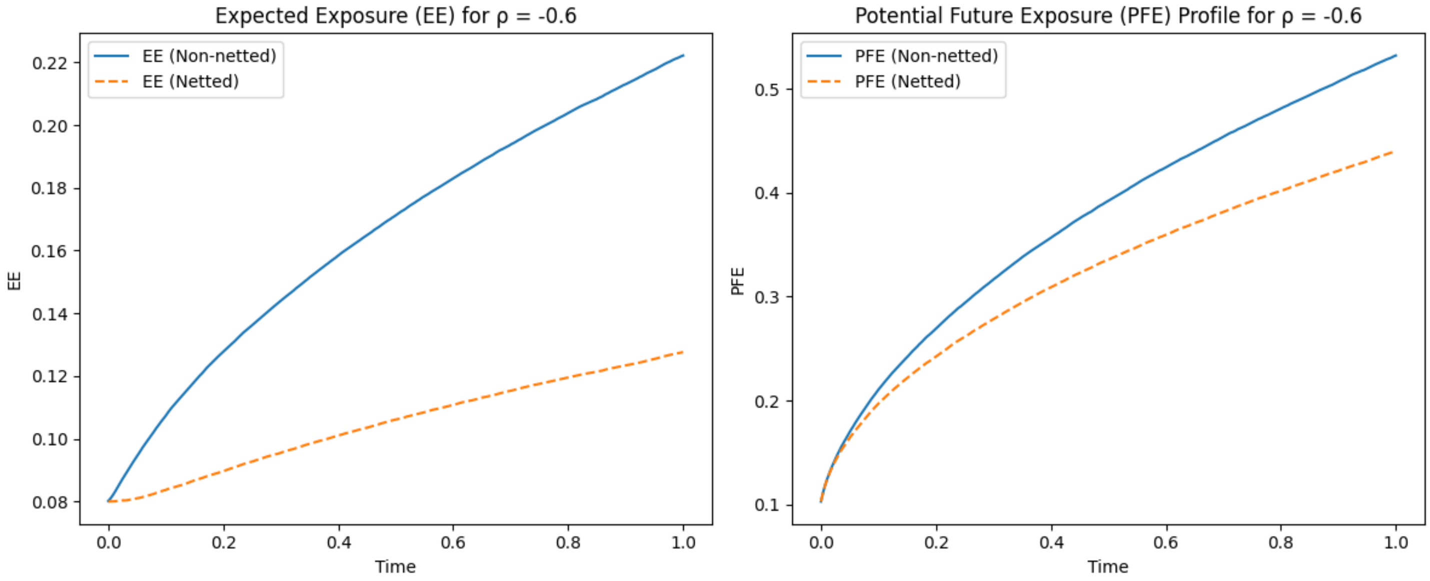
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
np.random.seed(1005778428)

# Parameters
mu1, mu2 = 0.05, 0.03  # expected returns for contracts 1 and 2
sigma1, sigma2 = 0.2, 0.25  # volatilities for contracts 1 and 2
T = 1  # time horizon in years
dt = 1/252  # time step (daily)
n_steps = int(T / dt)
n_simulations = 100000  # number of Monte Carlo simulations
correlations = [-0.99, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 0.99]  # different correlation values

# Function to generate correlated Wiener processes using Cholesky decomposition
def generate_correlated_wiener_processes(rho, n_simulations, n_steps):
    # Correlation matrix
    corr_matrix = np.array([[1, rho], [rho, 1]])
    # Cholesky decomposition
    L = np.linalg.cholesky(corr_matrix)

    # Generate independent standard normal samples
    Z = np.random.normal(0, np.sqrt(dt), (2, n_simulations, n_steps))

    # Apply the Cholesky decomposition to introduce correlation
    correlated_W = np.einsum('ij,jkl->ikl', L, Z)  # Matrix multiplication along correct dimensions

    # Cumulative sum to get Wiener processes
    W1, W2 = correlated_W[0].cumsum(axis=1), correlated_W[1].cumsum(axis=1)

    return W1, W2

# Function to calculate exposures
def calculate_exposures(rho):
    W1, W2 = generate_correlated_wiener_processes(rho, n_simulations, n_steps)
    MtM1 = mu1 + sigma1 * W1
    MtM2 = mu2 + sigma2 * W2
    # Non-netted exposure
    E_non_net = np.maximum(MtM1, 0) + np.maximum(MtM2, 0)
    EE_non_net = E_non_net.mean(axis=0)
    # Calculate PFE as the mean of the top 10% exposures at each time step
    PFE_non_net = np.array([np.mean(E_non_net[:, i][E_non_net[:, i] >=
                  np.percentile(E_non_net[:, i], 90)]) for i in range(n_steps)])

    # Netted exposure
    E_net = np.maximum(MtM1 + MtM2, 0)
    EE_net = E_net.mean(axis=0)
    # Calculate PFE as the mean of the top 10% exposures at each time step
    PFE_net = np.array([np.mean(E_net[:, i][E_net[:, i] >=
              np.percentile(E_net[:, i], 90)]) for i in range(n_steps)])

    return EE_non_net, PFE_non_net, EE_net, PFE_net

# Plotting function
def plot_exposures(EE_non_net, PFE_non_net, EE_net, PFE_net, rho):
    time_grid = np.linspace(0, T, n_steps)

    plt.figure(figsize=(12, 5))

    # Plot for Expected Exposure (EE)
    plt.subplot(1, 2, 1)
    plt.plot(time_grid, EE_non_net, label='EE (Non-netted)', linestyle='-', linewidth=1.5)
    plt.plot(time_grid, EE_net, label='EE (Netted)', linestyle='--', linewidth=1.5)
    plt.title(f'Expected Exposure (EE) for  = {rho}')
    plt.xlabel('Time')
    plt.ylabel('EE')
    plt.legend()

    # Plot for Potential Future Exposure (PFE)
    plt.subplot(1, 2, 2)
    plt.plot(time_grid, PFE_non_net, label='PFE (Non-netted)', linestyle='-', linewidth=1.5)
    plt.plot(time_grid, PFE_net, label='PFE (Netted)', linestyle='--', linewidth=1.5)
    plt.title(f'Potential Future Exposure (PFE) Profile for  = {rho}')
    plt.xlabel('Time')
    plt.ylabel('PFE')
    plt.legend()

    plt.tight_layout()
    plt.show()

# Run simulations, plot, and capture final values for each correlation
final_values = []

for rho in correlations:
    EE_non_net, PFE_non_net, EE_net, PFE_net = calculate_exposures(rho)
    plot_exposures(EE_non_net, PFE_non_net, EE_net, PFE_net, rho)

    final_EE_non_net = EE_non_net[-1]
    final_PFE_non_net = PFE_non_net[-1]
    final_EE_net = EE_net[-1]
    final_PFE_net = PFE_net[-1]

    final_values.append({
        'Correlation ()': rho,
        'Final EE (Non-netted)': final_EE_non_net,
        'Final PFE (Non-netted)': final_PFE_non_net,
        'Final EE (Netted)': final_EE_net,
        'Final PFE (Netted)': final_PFE_net
    })

# Create dataframe and round values to 4 decimal points
final_values_df = pd.DataFrame(final_values).round(4)

# Display final results
print(final_values_df)
```

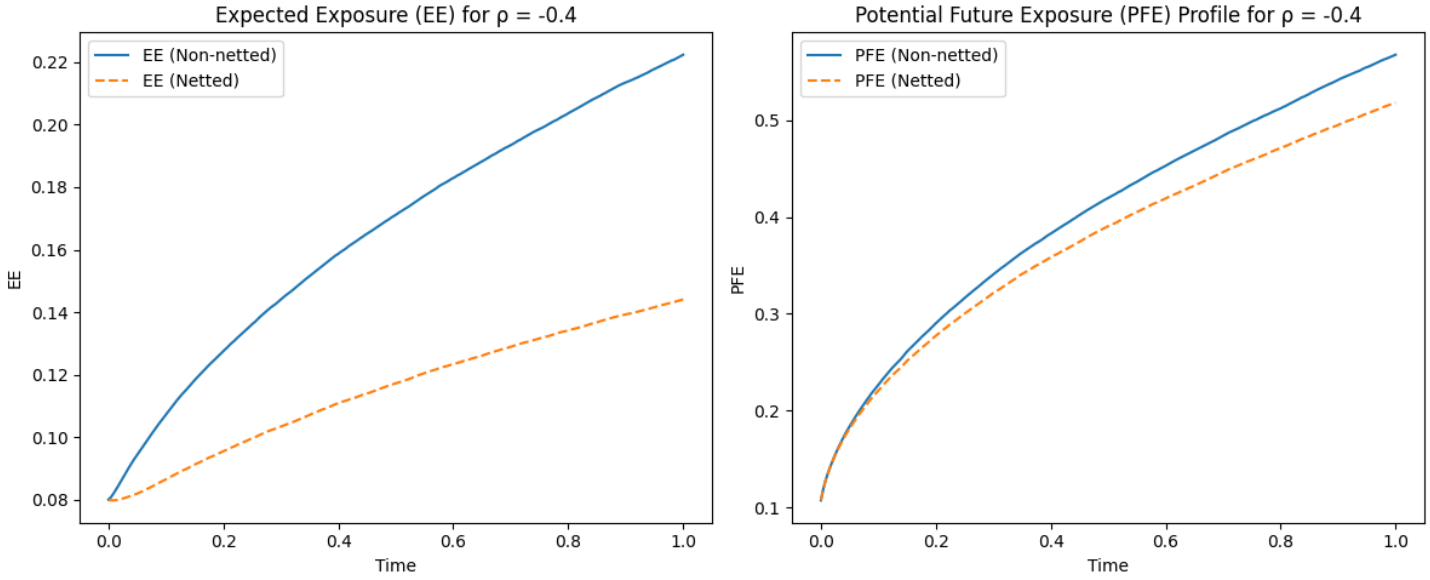


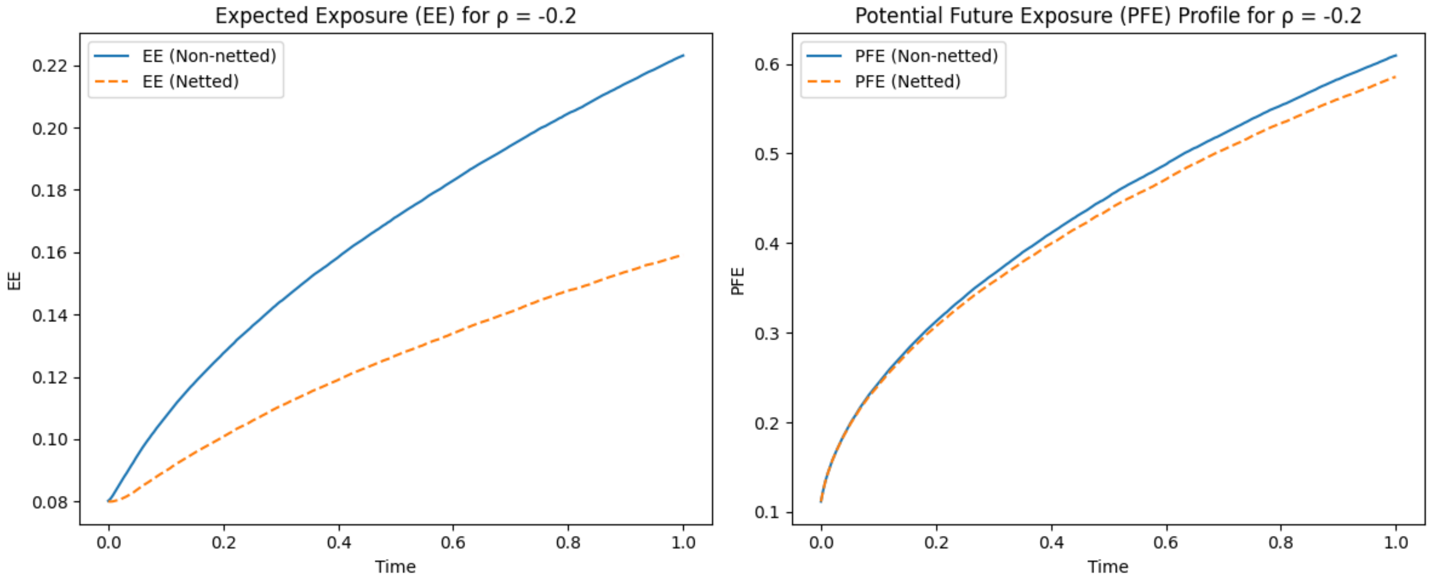


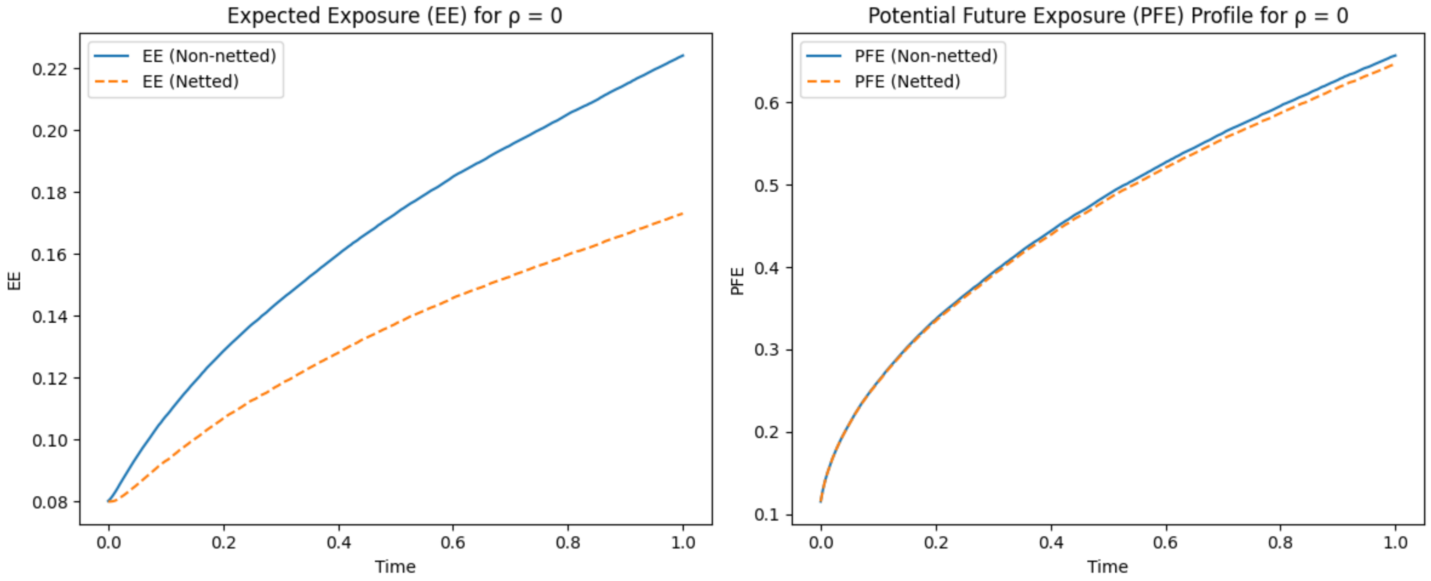


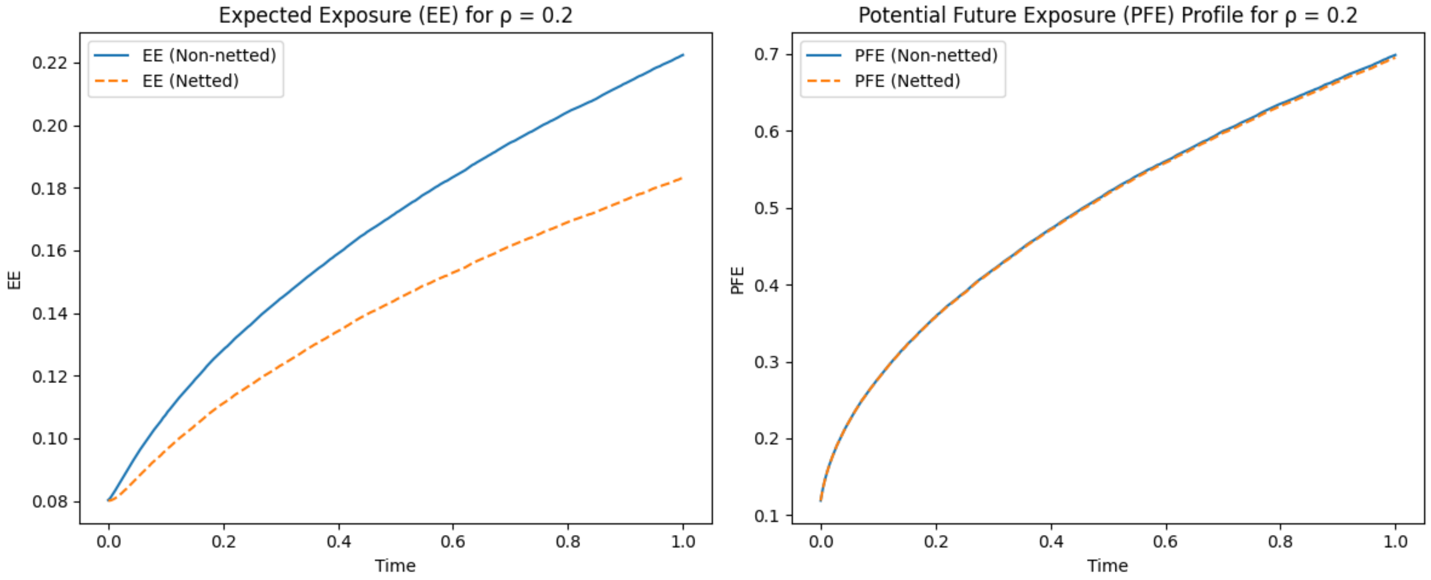


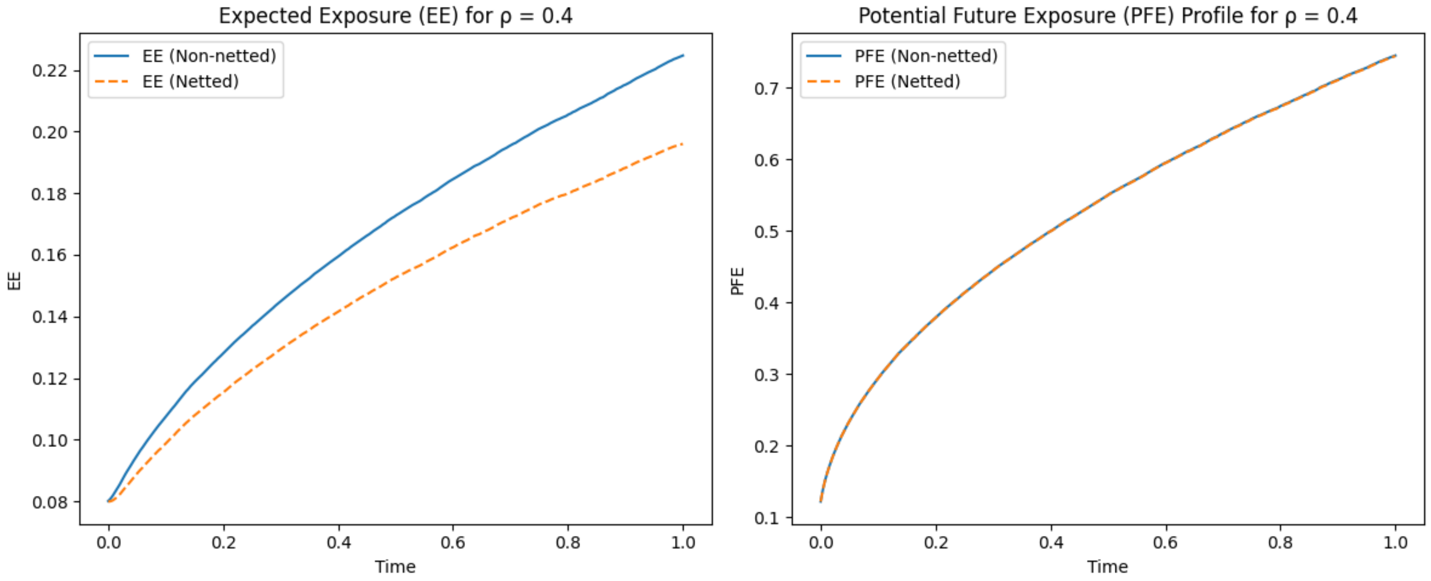

Expected Exposure (EE) for ρ = -0.6
EE (Non-netted)
EE (Netted)
EE
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = -0.6
PFE (Non-netted)
PFE (Netted)
PFE
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
Time

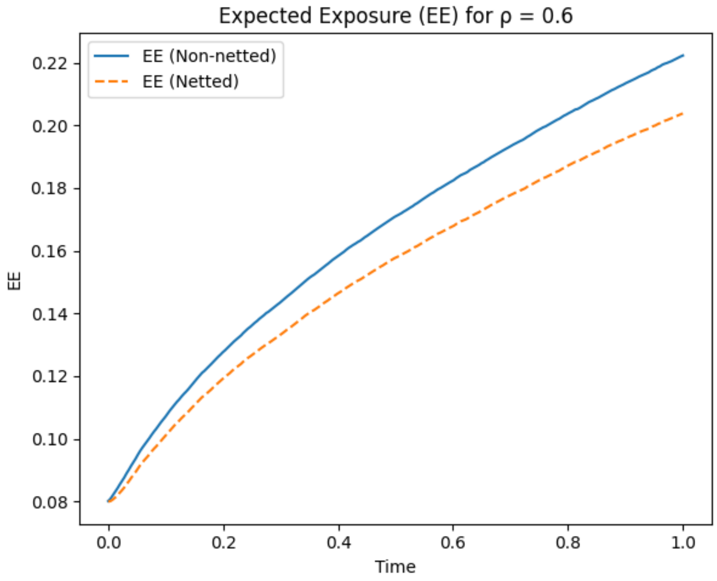

Expected Exposure (EE) for ρ = -0.4
EE (Non-netted)
EE (Netted)
EE
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = -0.4
PFE (Non-netted)
PFE (Netted)
PFE
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
Time


Expected Exposure (EE) for ρ = -0.2
EE (Non-netted)
EE (Netted)
EE
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = -0.2
PFE (Non-netted)
PFE (Netted)
PFE
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
Time

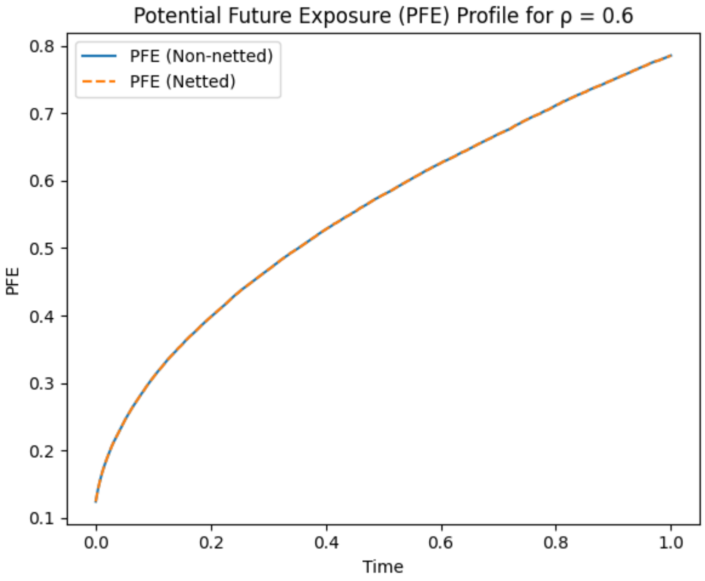

Expected Exposure (EE) for ρ = 0
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 0
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Time


Expected Exposure (EE) for ρ = 0.2
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 0.2
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8
1.0
Time

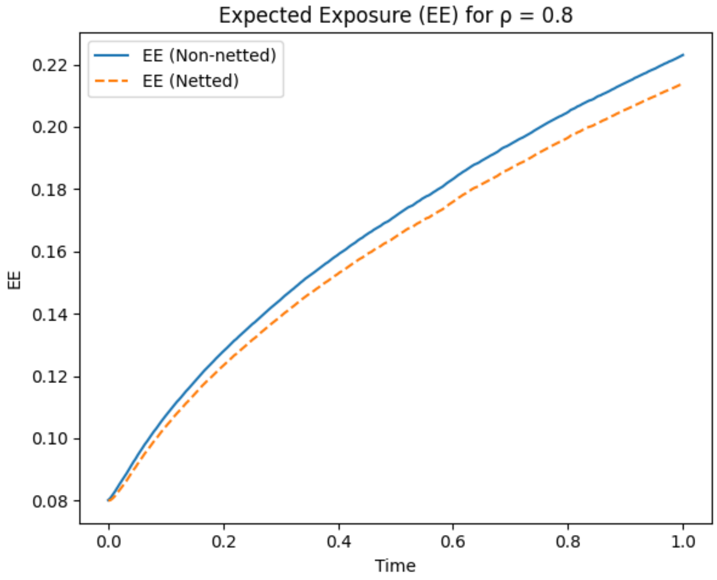

Expected Exposure (EE) for ρ = 0.4
EE (Non-netted)
EE (Netted)
EE
0.08
0.10
0.12
0.14
0.16
0.18
0.20
0.22
0.0
0.2
0.4
0.6
0.8
1.0
Time
Potential Future Exposure (PFE) Profile for ρ = 0.4
PFE (Non-netted)
PFE (Netted)
PFE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.0
0.2
0.4
0.6
0.8
1.0
Time



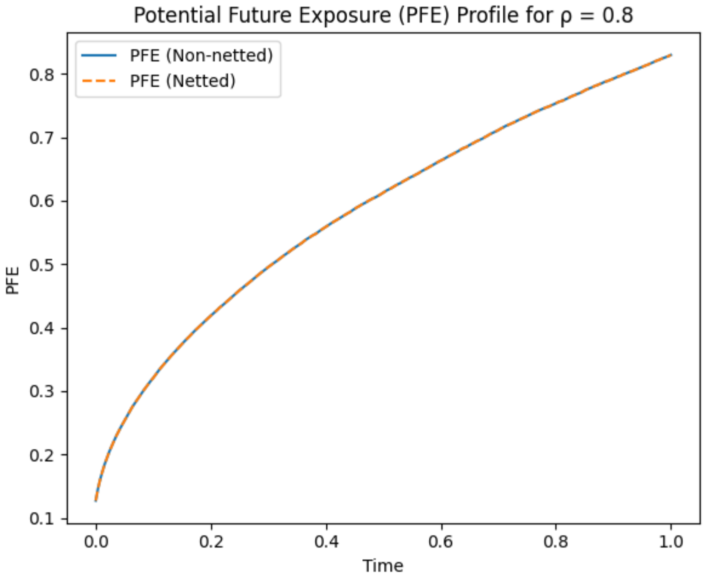
Expected Exposure (EE) for ρ = 0.6



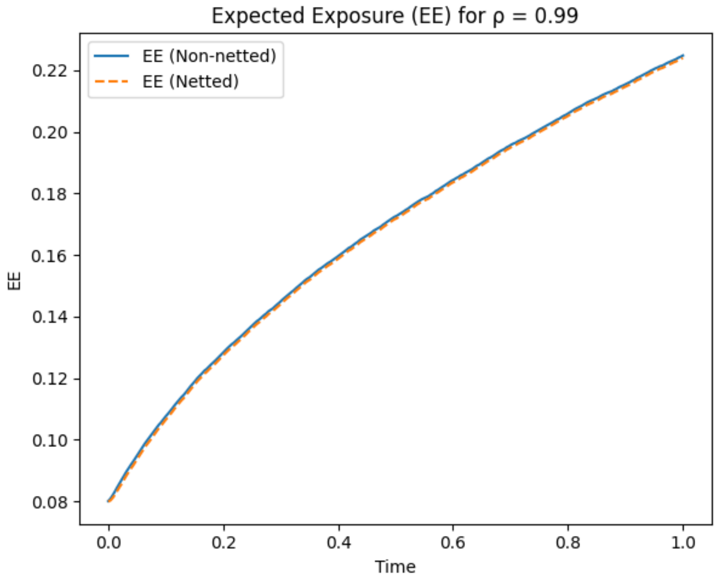
Potential Future Exposure (PFE) Profile for ρ = 0.6



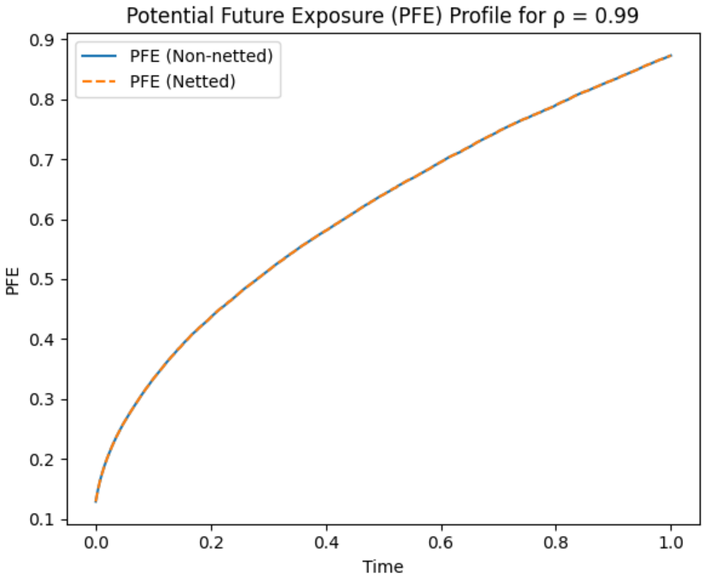
Expected Exposure (EE) for ρ = 0.8



Potential Future Exposure (PFE) Profile for ρ = 0.8



Expected Exposure (EE) for ρ = 0.99



Potential Future Exposure (PFE) Profile for ρ = 0.99

Compute non-netted and netted EE as well as non-netted and netted PFE at time $T = 1$:

Correlation (ρ)	Final EE (Non-netted)	Final EE (Netted)	Final PFE (Non-netted)	Final PFE (Netted)
-0.99	0.2226	0.0823	0.5086	0.1838
-0.80	0.2228	0.1081	0.5116	0.3429
-0.60	0.2222	0.1276	0.5321	0.4399
-0.40	0.2224	0.1441	0.5675	0.5179
-0.20	0.2232	0.1591	0.6092	0.5853
0.00	0.2240	0.1729	0.6565	0.6461
0.20	0.2224	0.1832	0.6988	0.6953
0.40	0.2247	0.1960	0.7454	0.7448
0.60	0.2223	0.2038	0.7854	0.7853
0.80	0.2231	0.2139	0.8299	0.8299
0.99	0.2248	0.2240	0.8727	0.8727

Table 2: Expected Exposure (EE) and Potential Future Exposure (PFE) values for different correlation levels.

Comments:

For expected exposure, the non-netted EE is consistently higher than the netted EE across all correlation values since netting reduces exposure by offsetting positive and negative positions. At $\rho = 0$, netting significantly reduces exposure, as the position offset each other perfectly in opposite directions. As correlation increases, netted EE approaches non-netted EE. At $\rho = 1$, netted EE and non-netted EE coincide.

For potential future exposure, the non-netted PFE is consistently higher than the netted PFE across all correlation values, similar to expected exposure. As correlation increases, netted PFE approaches non-netted PFE as well. At $\rho = 1$, netted PFE and non-netted PFE also coincide.

PFE profile is always higher than EE since PFE is a tail measure of exposure, capturing top 10% extreme outcomes, while EE represents an average exposure across scenarios.

In summary, in both EE and PFE cases, an increase in correlation reduces the effectiveness of netting.