Fin537 Homework 1 Solution

Group Memerbers:
Shicheng Zhang (sz90)
Chengjia Dong (dong47)
Junru Wang (junruw2)
Junhong Huang (jh136)

Question1

1. (a) $\Delta_{ABC} = 40000.1 - 80000.05 = 0$

$\Delta_{DEF} = 5000 \cdot |-0.55| = -2750$

(b) $\sigma_P = \sqrt{(0)^2(0.12)^2 + (-2750)^2(0.10)^2 + 2 \cdot 0} = 275$

VaR = 2750*0.01 - 1.645 * 275 = 479.875

(c) Option have a "Non-Linear Return" Characteristic.
So This method may ignore Gamma, Vega
and cause error when price change is big.

Question2

(a)(b)

2. (a) $\Delta = 1000$

(b) $VaR = \Delta \times F_6 \times 1000 = 46520$ (BCD)

(c)

The Delta-Normal method is likely to provide a correct estimate.
Because the simple rate of return on the BC stock index is normally distributed and index fund and the payoff of the index fund have a linear relationship with the index, which are aligned with the assumptions of the Delta-Normal method.

Question3

Q3

(a) $V_{FHD} = 8 \times 100000$

$= 800000$

$\Delta S = \frac{\partial V_{FHD}}{\partial S} = \frac{800000}{100}$

$= 8000$

(b) $V_{FHD} = x \cdot V_{USD}$

$\Delta x = \frac{\partial FHD}{\partial x} = V_{USD}$

$= 100000$

(c) $VAR = k \cdot \sigma_p$

$= k \cdot \sqrt{(\Delta_S \sigma_S)^2 + (\Delta_x \sigma_x)^2 + 2\rho \Delta_S \Delta_x \sigma_S \sigma_x}$

$= 2.326 \times \sqrt{16^2 + 5000^2 + 2 \cdot 0.5 \cdot 16 \cdot 5000}$

$= 2.326 \times \sqrt{25825856}$

$= 2.326 \times 5081.8894$

$= 11820.47$

(d) $\frac{VAR}{V_{FHD}} = \frac{11820.47}{800000} = 0.1478$

Question4

4. (a) $\Delta = 1000 - 20 \times 50 = 0$

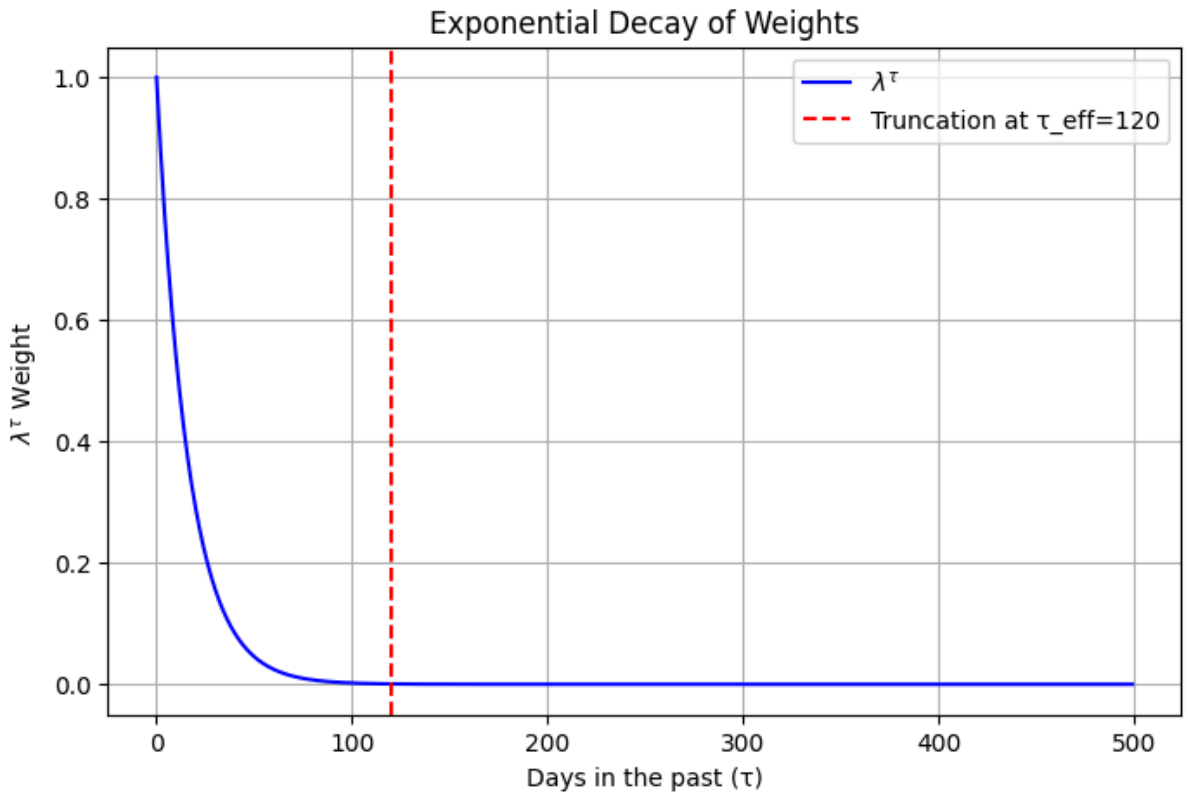
(b) $\frac{\partial V_{fund}}{\partial e}$ = fund value in FHD = 1,000,000

$\frac{\partial V_{future}}{\partial e} = 0$

$\frac{\partial V_p}{\partial e} = 1{,}000{,}000 + 0 = 1{,}000{,}000$

Question5

```python
import numpy as np
import matplotlib.pyplot as plt


# λ = 0.94
lambda_ewma = 0.94
weights = np.array([(1 - lambda_ewma) * lambda_ewma**i for i in range(len(returns_df))])
weights /= weights.sum()


#lambda^τ_eff < 0.0006
tau_eff = np.log(0.0006) / np.log(lambda_ewma)
tau_eff = int(np.ceil(tau_eff))

tau_values = np.arange(0, 500)
weights = lambda_ewma ** tau_values


plt.figure(figsize=(8, 5))
plt.plot(tau_values, weights, label=r'$\lambda^\tau$', color='blue')
plt.axvline(tau_eff, color='red', linestyle='--', label=f'Truncation at τ_eff={tau_eff}')
plt.xlabel('Days in the past (τ)')
plt.ylabel(r'$\lambda^\tau$ Weight')
plt.title('Exponential Decay of Weights')
plt.legend()
plt.grid(True)
plt.show()

tau_eff
```



Find best length of days is 120

```python
end_date = pd.Timestamp("2023-06-30")
returns_df = returns_df[returns_df['Date'] <= end_date].tail(120)

returns_df.sort_values('Date', inplace=True)

lambda_ewma = 0.94
tau_eff = 120

weights = np.array([(1 - lambda_ewma) * lambda_ewma**(tau_eff - 1 - i) for i in range(tau_eff)])
weights /= weights.sum()
returns_matrix = returns_df[etf_list].values

mean_returns = np.average(returns_matrix, axis=0, weights=weights)

deviations = returns_matrix - mean_returns

ewma_cov_matrix = np.dot(deviations.T * weights, deviations)

ewma_cov_df = pd.DataFrame(ewma_cov_matrix, index=etf_list, columns=etf_list)
ewma_cov_df
```

	XLB	XLE	XLF	XLP	XLV	XLY
XLB	0.000109	0.000092	0.000073	0.000036	0.000031	0.000045
XLE	0.000092	0.000166	0.000073	0.000022	0.000019	0.000004
XLF	0.000073	0.000073	0.000080	0.000021	0.000028	0.000048
XLP	0.000036	0.000022	0.000021	0.000036	0.000025	0.000025
XLV	0.000031	0.000019	0.000028	0.000025	0.000044	0.000029
XLY	0.000045	0.000004	0.000048	0.000025	0.000029	0.000117

Question6

```
portfolio_date = pd.Timestamp("2023-06-30")
prices_df = prices_df[prices_df['Date'] <= end_date].tail(120)
prices_at_date = prices_df[prices_df['Date'] == portfolio_date][etf_list].values.flatten()

holdings = np.array([10000, 10000, 20000, 15000, 10000, 10000])

p = prices_at_date * holdings
#Delta-Normal VaR
z_95 = 1.645
var_95 = z_95 * np.sqrt(p @ ewma_cov_matrix @ p.T)
var_95
```

```
69987.83913602598
```