MFE409 - Risk; HW2

Aliaksei Kanstantsinau

In [6]:
```python
import numpy as np
```

Problem 1.

a) Derive a formula for the expected shortfall if gains are normally distributed $N(μ, σ^2)$

Let the random variable X, follow a normal distribution: $$X \sim N(\mu,\sigma^2)$$

The Value at Risk at confidence level α: $$P(X \leq VaR_{\alpha}) = 1 - \alpha$$ For a normal distribution, VaR at level α can be expressed as: $$VaR_{\alpha} = \mu + \sigma Z_\alpha$$ where Z_α is the $(1-\alpha)$ - quantile of the standard normal distribution.

Expected Shortfall at confidence level α: $$ES_{\alpha} = E[X\mid X\leq VaR_{\alpha}]$$ Conditional expectation can be derived using the density function where ϕ is the standard normal probability density function: $$ES_{\alpha} = \mu + \sigma \frac{\phi(Z_\alpha)}{1-\alpha}$$

Given $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$, sub Z_α into the formula: $$ES_{\alpha} = \mu - \sigma \frac{\phi(Z_\alpha)}{\alpha}$$

Since $\phi(z)$ and Z_α correspond to the lower tail, the formula becomes: $$ES_{\alpha} = \mu - \sigma \frac{\phi(-Z_\alpha)}{\alpha}$$ However, $\phi(-z) = \phi(z)$, and hence, $$ES_{\alpha} = \mu - \sigma \frac{\phi(Z_\alpha)}{\alpha}$$

Thus, the formula for the Expected Shortfall when gains follow a normal distribution $N(\mu, \sigma^2)$ is: $$ES_{\alpha} = \mu - \sigma \frac{\phi(Z_{1-\alpha})}{\alpha}$$

b) The expected shortfall can also be defined as the average of the VaR for all con dence level above c: $$ES = \frac{1}{1 – c}\int_{c}^{1}VaR_{α}dα$$ Prove that this definition is equivalent to the one we have seen in class. Hint: You can use integration by part and a change of variable.

$$ ES = \frac{1}{1-c} \int_{c}^1 (\Phi^{-1} (1-\alpha) \sigma + \alpha) d\alpha$$$$ = \frac{1}{1-\alpha}(\int_{c}^1 \Phi^{-1}(1-\alpha)\sigma d\alpha + \int_{c}^1 \alpha d\alpha)$$$$ = \frac{1}{1-\alpha}(\int_{c}^1 \Phi^{-1}(1-\alpha)\sigma d\alpha + \alpha)$$ Perform substitution: $\alpha=\Phi(y)$, $y= \Phi^{-1}(\alpha)$, $d\alpha = \phi(y)dy$, $\phi(y)=\frac{1}{\sqrt{2\pi}} \exp \left(– \frac{y^2}{2} \right)$ and solve the integral

$$ \int_{c}^1 \Phi^{-1}(1-\alpha) \sigma du = \int_{\Phi^{-1}(c)}^{\Phi^{-1}(1)}\Phi^{-1}(1-\Phi(y))\sigma\Phi(y)dy$$$$= \int_{\Phi^{-1}(c)}^\infty -y \sigma \phi (y) dy$$$$= \int_{\Phi^{-1} (c)}^\infty -\frac{y \sigma}{\sqrt{2\pi}} \exp \left(– \frac{y^2}{2} \right) dy$$

$$= \frac{\sigma}{\sqrt{2\pi}} \int_{\Phi^{-1} (c)}^\infty -y \exp \left(– \frac{y^2}{2} \right) dy$$$$= \frac{\sigma}{\sqrt{2\pi}}(\exp(– \frac{y^2}{2})|_{\Phi^{-1} (c)}^\infty$$$$= \frac{\sigma}{\sqrt{2\pi}}(0 – \exp(– \frac{\Phi^{-1} (c)^2}{2}t))$$$$= -\sigma \frac{1}{\sqrt{2\pi}}\exp(– \frac{\Phi^{-1} (c)^2}{2})$$$$= -\sigma \phi(\Phi^{-1} (c)) $$

$$ES= \frac{1}{1-c}[-\sigma \phi (\Phi^{-1} (c))] + \alpha$$$$= \alpha – \sigma \frac{\phi(\Phi^{-1}(c))}{1-c}$$

Problem 2.

a) Assume you have a portfolio of \$3m in asset A, \$4m in asset B and \$3m in asset C. What is the VaR of your portfolio? What is the CVaR and DVaR for each of the assets? Check that the sum of CVaRs coincides with VaR. Which asset is responsible for the most risk of the portfolio?

- From my draw, asset C carries the most risk in the portfolio. CVaRs add up to VaR (code below)

In [7]:
```python
def draw_returns(N):
    # coin flips
    normal_year = np.random.binomial(1, 0.9, N)

    # draw for normal years
    mu = np.array([0.05, 0.05, 0.05])
    Sigma = np.array([[0.09, 0.012, 0.021], [0.012, 0.16, 0.028], [0.021, 0.
    normal_ret = np.random.multivariate_normal(mu, Sigma, N)

    # draws for special years
    mu = np.array([-0.1, -0.1, -0.1])
    Sigma = np.array([[0.36, 0.24, 0.42], [0.24, 0.64, 0.56], [0.42, 0.56, 1
    special_ret = np.random.multivariate_normal(mu, Sigma, N)

    # combine
    ret = normal_ret
    for i in range(N):
        if normal_year[i] == 0:
            ret[i,:] = special_ret[i,:]

    return(ret)
```

In [23]:
```python
# draw returns
returns = draw_returns(1)
# set my position
position = np.array([3000000, 4000000, 3000000])
```

```
# get portfolio returns
portfolio_returns = returns.dot(position)
# get var for 99%
alpha = 0.01
portfolio_var = np.percentile(portfolio_returns, 100 * alpha)
print(f'Portfolio VaR:', portfolio_var)
```

```
Portfolio VaR: 2685995.9576035817
```

In [43]:

```
def calculate_vars(positions):
    cvars = []
    dvars = []
    for i in range(len(positions)):
        # calculate portfolio returns excluding the current asset
        temp_positions = positions.copy()
        # set position to zero
        temp_positions[i] = 0
        # get new returns
        temp_portfolio_returns = returns.dot(temp_positions)
        # get var
        temp_portfolio_var = np.percentile(temp_portfolio_returns, 100 * alp
        # get dvar
        cvar = portfolio_var - temp_portfolio_var
        cvars.append(cvar)
        dvar = cvars[i]/portfolio_var
        dvars.append(dvar)
    print(f'CVaR for each asset:', cvars)
    print(f'DVaR for each asset', dvars)

calculate_vars(position)
```

```
CVaR for each asset: [789785.750294473, 381166.96829852276, 1515043.23901058
62]
DVaR for each asset [0.29403832424197385, 0.14190898806809674, 0.56405268768
99294]
```

b) When you approximate the derivatives involved in DVaR and CVaR, vary the size of the position change you use. What do you observe when you change the value? Report a graph that shows the effect of different sizes of the step for the derivative. Explain what is happening.

- it doesnt look like my code yields correct answer.

In [52]:

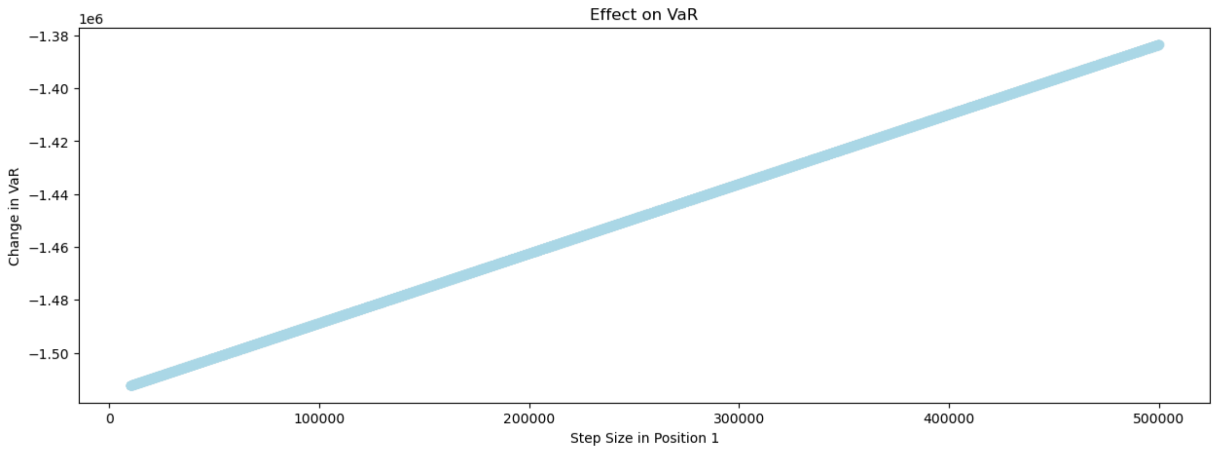
```
#import matplotlib.pyplot as plt

# define function that computes var
#def compute_var(returns, alpha=0.01):
    #return np.percentile(returns, 100 * alpha)

# set steps
#step_sizes = np.linspace(10000, 500000, 5000)
#var_changes = []

# compute VaR for each step size
#for step in step_sizes:
    #new_positions = positions + np.array([step, 0, 0])  # change only the f
    #new_portfolio_returns = returns.dot(new_positions)
    #new_var = compute_var(new_portfolio_returns)
    #var_change = new_var - portfolio_var
    #var_changes.append(var_change)

# plot results
#plt.figure(figsize=(15, 5))
#plt.plot(step_sizes, var_changes, marker='o', color='lightblue')
#plt.title('Effect on VaR')
#plt.xlabel('Step Size in Position 1')
#plt.ylabel('Change in VaR')
#plt.show()
```



c) You change your portfolio to \$3m in asset A, \$5m in asset B and \$3m in asset C. What are the CVaR and DVaR for asset C?

In [44]:
```
position2 = np.array([3000000, 5000000, 3000000])
calculate_vars(position2)
```

```
CVaR for each asset: [694494.0082198423, 381166.96829852276, 1419751.4969359
555]
DVaR for each asset [0.2585610772249497, 0.14190898806809674, 0.528575440672
9053]
```

Problem 3.

Deutsche Bank (DB) is a German bank that manages its book in EUR. Consider 2 desks in DB, one is long 150 million USD and the other is short 50 million GBP. The exchange rates are 1 USD = 0.93 EUR and 1 GBP = 1.17 EUR. The daily volatilities for changes in USD/EUR and GBP/EUR are 0.40% and 0.30%, respectively and means of 1 basis point and 2 basis points. The correlation between them is 0.7. For risk calculations, assume that the returns have mean zero and are normally distributed.
a) What is the 99% 1-day VaR for each desk?

In [17]:
```
# get eur value for desk 1
desk1 = 150000000*.93
# get mean for desk 1 assuming 1bp mean
desk1_mu = desk1 * .0001
```

```
# get eur value for desk 2
desk2 = 50000000*1.17
# get mean for desk 2 assuming 2bp mean
desk2_mu = desk2 * .0002
# get var for long desk
var_long = (desk1_mu - 2.32 * .4)
var_short = (desk2_mu - 2.32 * .3)
print(f'VaR for long position:', var_long)
print(f'VaR for short position:', var_short)
```

```
VaR for long position: 13949.072
VaR for short position: 11699.304
```

b) What is the 99% 1-day VaR for the combined portfolio?

In [18]:
```
var_combined = np.sqrt(var_long*var_short*.7)
print(f'VaR for both desks:', var_combined)
```

```
VaR for both desks: 10688.129101583756
```