# Q1.Expected shortfall

1. Derive a formula for the expected shortfall if gains are normally distributed $N(\mu, \sigma^2)$

set condifence level $c s.t. VaR = W_0 - Z_{1-c} = W_0 - F^{-1}(1-c)$

$$W \sim N(\mu, \sigma^2), f(W) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(W-\mu)^2}{2\sigma^2}}, F(c) = \int_{-\infty}^{c} f(W)dW$$

$$ES = W_0 - \frac{\int_{-\infty}^{W_0 - VaR} W f(W)dW}{\int_{-\infty}^{W_0 - VaR} f(W)dW} = W_0 - \mu + \frac{\sigma}{\sqrt{2\pi}} \frac{e^{-\frac{(W_0 - VaR - \mu)^2}{2\sigma^2}}}{1-c}$$

2. The expected shortfall can also be defined as the average of the VaR for all confidence level above c:

$$ES = \frac{1}{1-c}\int_c^1 VaR_\alpha d\alpha$$

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.

$$VaR_\alpha = W_0 - Z_{1-\alpha} = W_0 - F^{-1}(1-\alpha)$$

$$\frac{1}{1-c}\int_c^1 VaR_\alpha d\alpha = W_0 - \frac{1}{1-c}\int_c^1 F^{-1}(1-\alpha)d\alpha$$

$$= W_0 - \frac{1}{1-c}\int_{-\infty}^{F^{-1}(1-c)} y dF(y) = W_0 - \frac{1}{1-c}\int_{-\infty}^{F^{-1}(1-c)} y\text{-}$$

$$= W_0 - \mu - \frac{\sigma}{1-c}\frac{1}{\sqrt{2\pi}} e^{-\frac{(F^{-1}(1-c)-\mu)^2}{2\sigma^2}}$$

$$= W_0 - \mu - \frac{\sigma}{1-c}\frac{1}{\sqrt{2\pi}} e^{-\frac{(W_0 - VaR - \mu)^2}{2\sigma^2}} = ES$$

# Q2. Decomposing the VaR of a portfolio

The attached python-code contains a function called “draw_returns” with one argument N. Calling the function returns N random draws for the returns of 3 assets, A, B, and C. Please take this function as given and do not modify it (assume it is a black box ).

1. 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?

In [2]:
```
# This function creates N draws from some three random variables
import numpy as np

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,
    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
    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 [13]:
```
N = 100000
c = 0.99 # confidence_level
returns = draw_returns(N)
def VaRs(weights, epsilon):
    portfolio_returns = np.dot(returns, weights)
    VaR = np.percentile(-portfolio_returns, 100*c)

    DVaRs = np.zeros(3)
    for i in range(3):
        new_weights = weights.copy()
        new_weights[i] += epsilon # increase weight of asset i
        new_portfolio_returns = np.dot(returns, new_weights)
        DVaRs[i] = (np.percentile(-new_portfolio_returns, 100*c) - VaR)/e

    CVaRs = weights * DVaRs
    return VaR,DVaRs,CVaRs

weights = np.array([3*10**6, 4*10**6, 3*10**6])
epsilon = 1 # increase $1
VaR,DVaRs,CVaRs = VaRs(weights, epsilon)
```

```
print("VaR:", VaR)
print("DVaR:", DVaRs)
print("CVaR:", CVaRs)
print("The sum of CVaRs is ", CVaRs.sum(), "which equals VaR.")
print("The asset B is responsible for the most risk of the portfolio.")
```

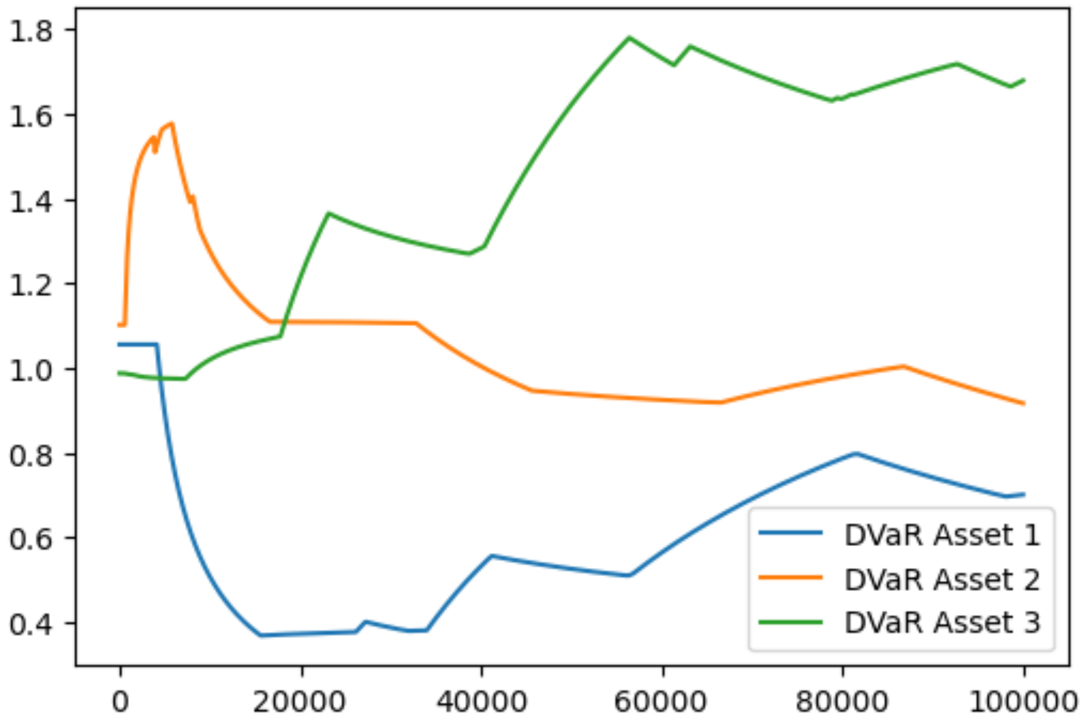
```
VaR: 10532470.975460406
DVaR: [1.05518242 1.10145048 0.9870406 ]
CVaR: [3165547.253564   4405801.91463232 2961121.79942429]
The sum of CVaRs is  10532470.967620611 which equals VaR.
The asset B is responsible for the most risk of the portfolio.
```

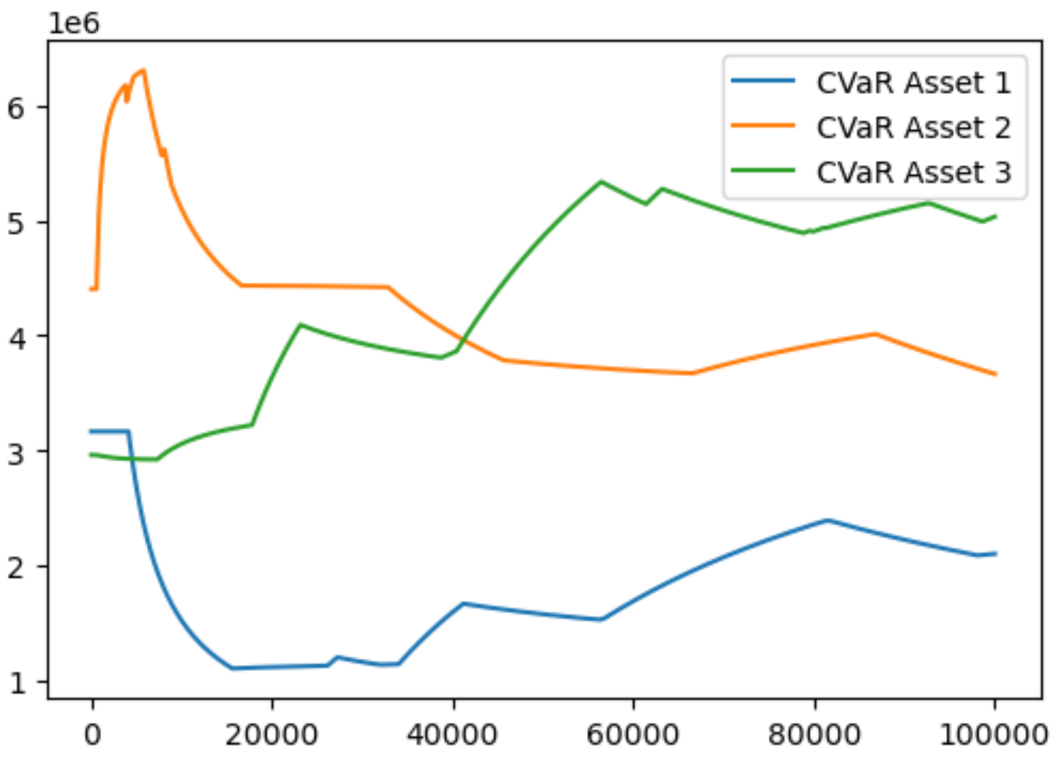
2. When you approximate the derivatives involved in DVaR and CVaR, vary the size of the position change you use. What do you observe when you change the value? Report a graph that shows the effect of different size of the step for the derivative. Explain what is happening.

In [14]:
```
import matplotlib.pyplot as plt
DVaR_list = []
CVaR_list = []
epsilon_list = np.arange(1,10**5,10)
for epsilon in epsilon_list:
    VaR,DVaRs,CVaRs = VaRs(weights, epsilon)
    DVaR_list.append(DVaRs)
    CVaR_list.append(CVaRs)

plt.figure(figsize=(6, 4))
for i, DVaR in enumerate(zip(*DVaR_list)):  # This unzips the list of lis
    plt.plot(epsilon_list, DVaR, label=f'DVaR Asset {i+1}')
plt.legend()
plt.figure(figsize=(6, 4))
for i, CVaR in enumerate(zip(*CVaR_list)):  # This unzips the list of lis
    plt.plot(epsilon_list, CVaR, label=f'CVaR Asset {i+1}')
plt.legend()
```

Out[14]:
```
<matplotlib.legend.Legend at 0x112363b10>
```




- Changes in DVaR show the sensitivity of an asset to changes in overall risk levels. These fluctuations may indicate that an asset's risk characteristics (such as volatility and inter-asset correlation) are very sensitive to changes in positions. For example, the DVaR of Asset 3 is much higher than that of Assets 1 and 2 at certain points, either because Asset 3 is relatively more volatile or because of its high correlation with other assets, resulting in the possibility that a small change in position could cause a relatively large change in VaR.

- CVaR changes indicate loss expectations in extreme situations.CVaR is more volatile compared to the DVaR chart due to the fact that CVaR quantifies tail risk, i.e., the average loss in the event of losses exceeding the conventional expectations. When CVaR changes significantly, it may indicate that certain assets may be exposed to larger extreme losses even with small position changes.

3. 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 [75]:
```
weights = np.array([3*10**6, 5*10**6, 3*10**6])
epsilon = 1 # increase $1
VaR,DVaRs,CVaRs = VaRs(weights, epsilon)
print("VaR:", VaR)
print("DVaR:", DVaRs)
print("CVaR:", CVaRs)
```

```
VaR: 11893478.4005178
DVaR: [0.20228331 1.23055733 1.71128062]
CVaR: [ 606849.91814196 6152786.64976358 5133841.85358882]
```

## Q3. Managing a Currency Trading Desk

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.

1. What is the 99% 1-day VaR for each desk?

$$
\begin{aligned}
c &= 0.99, z(c) = -2.326 \\
VaR &= -(\mu + \sigma \times z(c)) \\
\mu_{USD} &= 150 * 10^6 * 0.93 * 0.0001, \sigma_{USD} = 150 * 10^6 * 0.93 * 0.004 \\
VaR_{USD} &= -150 * 10^6 * 0.93 * (0.0001 - 0.004 * 2.326) = 1283958 EUR \\
\mu_{GBP} &= -50 * 10^6 * 1.17 * 0.0002, \sigma_{GBP} = 50 * 10^6 * 1.17 * 0.003 \\
VaR_{GBP} &= 50 * 10^6 * 1.17 * (0.0002 + 0.003 * 2.326) = 419913 EUR
\end{aligned}
$$

2. What is the 99% 1-day VaR for the combined portfolio?

$$\mu_{combined} = \mu_{USD} + \mu_{GBP} = 150 * 10^6 * 0.93 * 0.0001 - 50 * 10^6 * 1.17 * 0.000$$
$$\sigma_{combined} = \sqrt{\sigma_{USD}^2 + \sigma_{GBP}^2 + 2 \times correlation \times \sigma_{USD} \times \sigma_{GBP}} = 452839.541$$
$$\begin{aligned} VaR_{combined} &= -(\mu_{combined} + \sigma_{combined} \times z(c)) \\ &= -(2250 - 452839.5411 * 2.326) = 1051054.7726 EUR \end{aligned}$$

3. Optional, more challenging: Consider an arbitrary portfolio with positions $x_1$ and $x_2$ in two assets. If you increase your position in $x_1$ by a small amount $\triangle x_1$, by how much do you need to change your position in asset 2 to keep your VaR constant? What is the effect of these changes on your expected profits? Obtain mathematical expressions as function of (some but not neccesarily all of) $\triangle x_1$, VaR, DVaR, CVaR, and expected returns.

$$\begin{aligned} DVaR_i &= \frac{\partial Var}{\partial x_i}, CVaR_i = x_i \frac{\partial Var}{\partial x_i} \\ VaR &= \sum CVaR_i = x_1 \frac{\partial Var}{\partial x_1} + x_2 \frac{\partial Var}{\partial x_2} \\ &= (x_1 + \triangle x_1) \frac{\partial Var}{\partial x_1} + (x_2 + \triangle x_2) \frac{\partial Var}{\partial x_2} \\ \Rightarrow \triangle x_2 &= -\triangle x_1 \frac{\partial Var}{\partial x_1} / \frac{\partial Var}{\partial x_2} = -\triangle x_1 \frac{DVaR_1}{DVaR_2} \\ \triangle expected\_profit &= \triangle x_1 \mu_1 + \triangle x_2 \mu_2 \\ &= \triangle x_1 \mu_1 - \triangle x_1 \frac{DVaR_1}{DVaR_2} \mu_2 \\ &= \triangle x_1 (\mu_1 - \frac{DVaR_1}{DVaR_2} \mu_2) \end{aligned}$$

4. How would you change the allocation of DB's trading desk? Give a quantitative argument.

- If $\mu_1 - \frac{DVaR_1}{DVaR_2}\mu_2 > 0$, we can increase our expected profit by increasing our position in $x_1$ by $\triangle x_1$ and decrease our position in $x_2$ by $\triangle x_2$ without changing our VaR.