Q1. VaR for an exponential distribution

1. Assume an asset value in 10 days follows an exponential distribution with mean W_0. Derive a formula for the value-at-risk for confidence c and reference level W_0 for the asset. Apply with $W_0 = 50$ and $c = 98\%$.

$$W \sim Exp(\lambda), \lambda = 1/W_0, f(x) = \lambda e^{-\lambda x}, x \geq 0$$

$$F(x) = 1 - e^{-\lambda x}, x \geq 0, F^{-1}(y) = -\frac{ln(1-y)}{\lambda}$$

$$VaR = W_0 - F^{-1}(1-c) = W_0[1 + ln(c)] = 48.9898$$

$$ES = W_0 - \frac{\int_{-\infty}^{W_0 - VaR} Wf(W)dW}{\int_{-\infty}^{W_0 - VaR} f(W)dW} = W_0 - \frac{-(2W_0 - VaR)e^{-\frac{W_0 - VaR}{W_0}} + W_0}{1 - e^{-\frac{W_0 - VaR}{W_0}}}$$
$$= 49.4966$$

2. Now assume you are short this asset. Compute your value-at-risk for

confidence c and apply with the same numerical values.

$$VaR = F^{-1}(c) - W_0 = -W_0(ln(1-c) + 1) = 145.6011$$

$$ES = \frac{\int_{W_0+VaR}^{+\infty} Wf(W)dW}{\int_{W_0+VaR}^{+\infty} f(W)dW} - W_0 = W_0 + VaR = 195.6011$$

3. Comment on the similarity or difference between the results in the two

questions.

The VaR is larger when we short the asset and the shortfall is smaller when we short the asset.

5. Repeat the previous questions with expected shortfall.

Q2. VaR for mixtures

You have to allocate $\$1$bn in the stock market. You are discussing with your partner regarding the volatility of returns. She has a view that, in line with historical averages, the volatility of returns will be of $\sigma_1 = 12\%$ in the next year. However, you believe that volatility will be higher, in the orders of $\sigma_2 = 16\%$ for the next year. After discussing with your partner, you agree in the following way: stocks returns follow a normal distribution with mean μ and σ_1 with probability π and normal distribution with mean μ and σ_2 with probability $1-\pi$. For now assume that $\mu = 7\%$ and $\pi = 0.4$.

1. Compute the 1-year 99% VaR with your view, with your partner view, and with the common view. Compare results and provide a very brief explanation as if you were presenting to your manager.

In [37]:
```
from scipy.stats import norm
mu = 0.07
pi = 0.4
sigma_1 = 0.12
sigma_2 = 0.16
c = 0.99 # confidence_level

print("In partner view, VaR = ", -(mu+norm.ppf(1-c)*sigma_1))
print("In my view, VaR = ", -(mu+norm.ppf(1-c)*sigma_2))
```

```
In partner view, VaR =  0.20916174488490086
In my view, VaR =  0.30221565984653453
```

In [60]:
```
import numpy as np
def simulate(n_simulations, pi, mu, sigma_1, sigma_2):
    # Simulating returns
    returns_1 = np.random.normal(mu, sigma_1, int(n_simulations * pi))
    returns_2 = np.random.normal(mu, sigma_2, int(n_simulations * (1 - pi
    returns = np.concatenate((returns_1, returns_2))

    # Calculating VaR
    sorted_returns = np.sort(returns)
    var_index = int((1 - c) * n_simulations)
    var = sorted_returns[var_index]
    return -var

print("In common view, VaR = ", simulate(100000, pi, mu, sigma_1, sigma_2
```

```
In common view, VaR =  0.2742792498413824
```

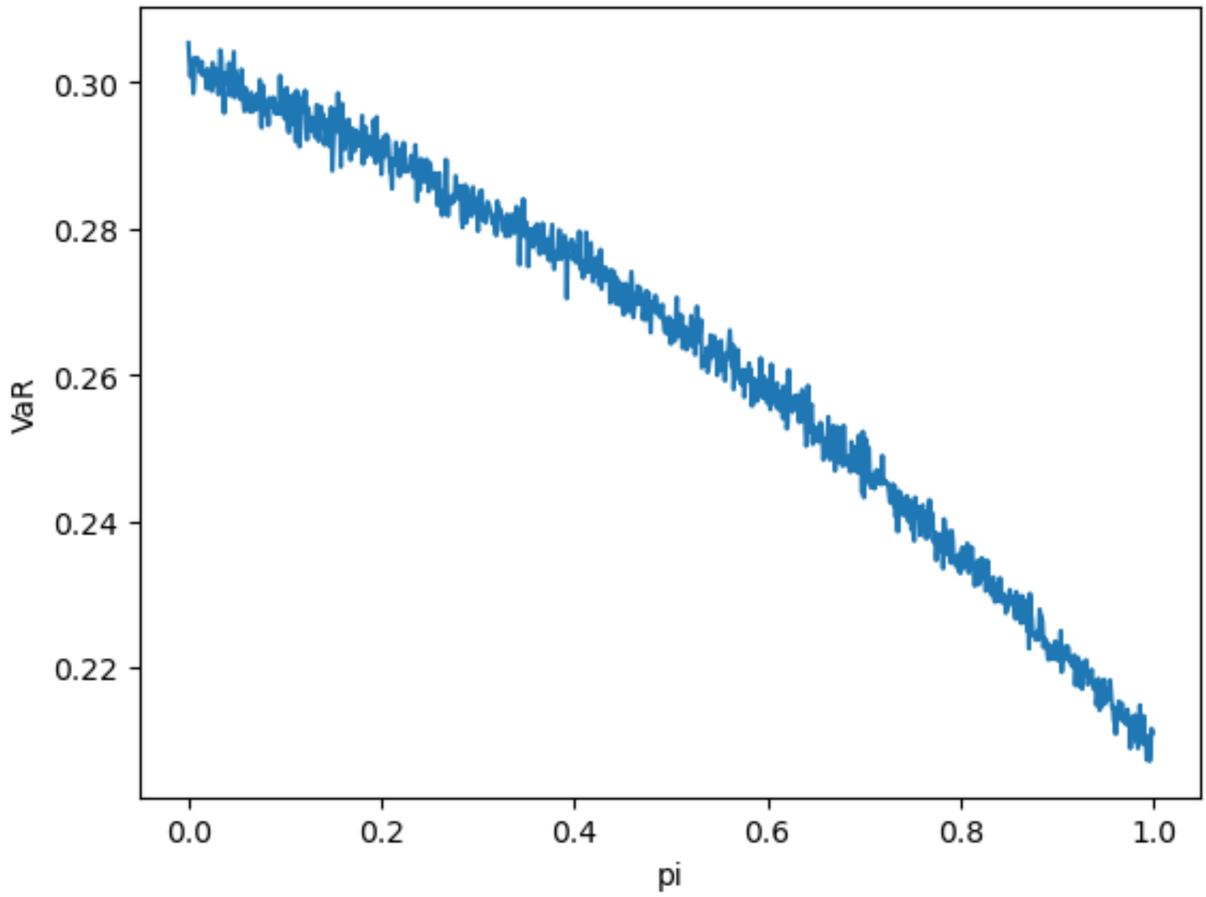
In a normal distribution, the larger the sigma (standard deviation), the higher the tails on both sides of the distribution curve. Consequently, for the same confidence level, the Value at Risk (VaR) becomes larger.

2. To understand the role of π, plot a chart with π between 0 and 1 on the horizontal axis and the corresponding VaR on the vertical axis. Comment on your results.

In [51]:
```
import matplotlib.pyplot as plt
pi_list = np.arange(0,1,0.001)
VaR_list = [simulate(100000, i, mu, sigma_1, sigma_2) for i in pi_list]
plt.plot(pi_list, VaR_list)
plt.xlabel('pi')
```

```
plt.ylabel('VaR')
```

Out[51]: Text(0, 0.5, 'VaR')



3. More challenging. After presenting your common view to your manager, you are challenged with an alternative view about volatility: σ is time-varying. The volatility trader suggests that a sensible model for σ is a gamma distribution. Explain in as many details as possible (either derive of formula or use a computer program) how to compute the VaR of your portfolio when returns have a normal distribution conditional on σ and σ is distributed according to a Gamma distribution.

$$\sigma \sim Gamma(\alpha, \theta)$$

In [87]:
```
from scipy.stats import gamma
alpha = 1
theta = 2
n_simulations = 1000000

sigmas = gamma.rvs(alpha, theta, size=n_simulations) * 0.05
returns = norm.rvs(mu, sigmas)
sorted_returns = np.sort(returns)
var_index = int((1 - c) * n_simulations)
var = sorted_returns[var_index]
```

```
-var
```

Out[87]: 0.3333768288319987

In [ ]: