

STK4505 - Mandatory Assignment 1

October 2025

1 Problems

Problem 1. A fixed-coupon bond is a financial instrument characterised by maturity T -years, coupon c (which is assumed to be paid out annually) and principal P . For such a bond, a bond holder gets cP amount of money at the end of each year before maturity, and at the maturity they get $cP + P$. The value of the bond at time $t = 0$ is equal to the present value of the underlying cashflows.

1. Assume that the interest rate at time $t = 0$ is flat and equal r . Write down a formula for calculating the value of the bond at time $t = 0$ given the parameters r, T, c, P .
2. Assume that $r = c$. Prove that the value of the bond is equal to P regardless of c, P, T .
3. Write a function which calculates the value of the bond, given parameters r, c, P, T . Run the function for $r = 5\%$, $c = 4\%$, $T = 10$ and $P = 1000$. Run the function again changing different parameters and plot the results (parameter against the value of the bond), first with the interest rate ($r = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\%$), then coupon ($c = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\%$) and at last, the maturity of the bond ($T = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20$). What does the results tell you about the relation between different level of parameters and the value of the bond? Why are those values different?
4. Calculate the duration of the bond for $r = 5\%$, $c = 4\%$, $T = 10$ and $P = 1000$ and give an interpretation to it. Assume that the interest rate has increased from $r = 5\%$ to $r = 6\%$. How much would the price of the bond change, as prescribed by the duration? What is the real change in the value of the bond? Why are those two values different?

Solution 1.

1. The formula for calculating the value of the bond at $t = 0$, given r, T, c, P is as follows:

$$V(r, T, c, P)(0) = \sum_{i=1}^T \frac{cP}{(1+r)^i} + \frac{P}{(1+r)^T}. \quad (1)$$

2. If $r = c$, the value of the bond given by (1) is

$$\begin{aligned}
 V(c, T, c, P)(0) &= \sum_{i=1}^T \frac{cP}{(1+c)^i} + \frac{P}{(1+c)^T} && \text{(geometric sum)} \\
 &= cP \left(\frac{1 - \frac{1}{(1+c)^{T+1}}}{1 - \frac{1}{1+c}} - 1 \right) + \frac{P}{(1+c)^T} \\
 &= cP \left(\frac{1 + c - \frac{1}{(1+c)^T}}{c} - 1 \right) + \frac{P}{(1+c)^T} \\
 &= P(1+c) - \frac{P}{(1+c)^T} - cP + \frac{P}{(1+c)^T} \\
 &= P
 \end{aligned}$$

3. The function can be found in section 2.1 in the Appendix. For $r = 5\%$, $c = 4\%$, $T = 10$ and $P = 1000$, it returns 922,7826507081516. The plots of the value function with respect to the different parameters are seen in Figure 1. As you can see, the price of the bond decreases with

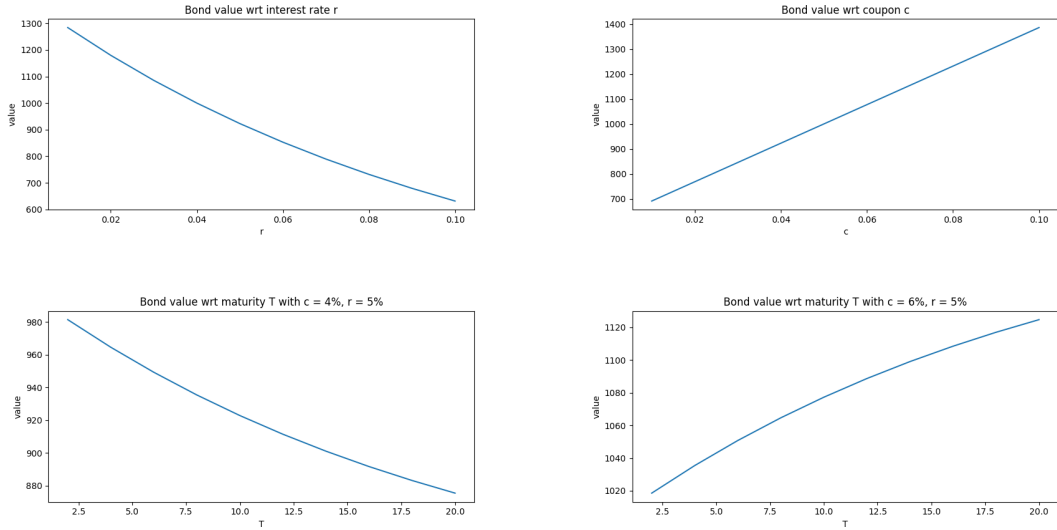


Figure 1: Bond values plotted w.r.t. interest rate r , coupon c and maturity T , respectively.

larger r and increases with larger c . With a large r , future payments are discounted more heavily, and a bond with a higher coupon is naturally more attractive. For increasing maturity T , we observe the bond price exhibiting two different behaviors: If $c < r$, it is decreasing, and if $c > r$, it is increasing. The reason for this can be seen by computing $V(r, T, c, P)(0)$ the same way as in

1.2:

$$\begin{aligned} V(r, T, c, P)(0) &= \sum_{i=1}^T \frac{cP}{(1+r)^i} + \frac{P}{(1+r)^T} \\ &= cP \left(\frac{1+r - \frac{1}{(1+r)^T}}{r} - 1 \right) + \frac{P}{(1+r)^T} \\ &= \frac{P}{r} \left(c + \frac{r-c}{(1+r)^T} \right). \end{aligned}$$

We can see that the last term inside the parentheses diminishes with T increasing. It is positive for $r > c$, so the bond value then decreases. On the other hand, if $c > r$, the last term is negative and V increases with T .

4. Let $r = 5\%$, $c = 4\%$, $T = 10$ and $P = 1000$. The duration of this bond is given by

$$D(r, T, c, P) = \frac{1}{V(r, T, c, P)} \left(\sum_{i=1}^T \frac{cPi}{(1+r)^i} + \frac{TP}{(1+r)^T} \right) \approx 8,36 \text{ years.}$$

This tells us the average time it takes to receive all the cash flows of a bond, weighted by the present value of each of the cash flows. The price change of the bond prescribed by the duration is

$$\Delta V_0 \approx -\frac{D}{1+r} V_0 \Delta r,$$

which equates to $\approx -73,47$ for a change $\Delta r = 0.01$ with r, c, T, P . However, the real change in price is

$$V(0,06, 10, 0,04, 1000)(0) - V(0,05, 10, 0,04, 1000)(0) \approx -69.98.$$

Those two values are different because the first approximation assumes that V varies linearly in r around that point.

Problem 2. We assume that a random sample $Z_1, \dots, Z_n$ is Rayleigh-distributed.

1. Derive an inverse sampler for the Rayleigh-distribution.
2. Derive the moment estimator of σ .
3. Write a log-likelihood function $L(\sigma)$. Derive the maximum likelihood estimator of σ .
4. Write a script for a function which calculates the maximum likelihood estimators given a sample $Z_1, \dots, Z_n$. The function should be general enough to handle different n .
5. Generate $n = 10, 100, 1000, 10000, 1000000$ simulations using the inverse sampler (simulations for different n should be drawn independently) for parameters $\sigma = 3$. Compute a moment estimator σ_{MM} and maximum likelihood estimator σ_{ML} for each n using a function from the previous points and prepare a table presenting the estimated parameters for different n . Compare it to the true value of parameter and comment on the estimation error.

6. Using a closed-form formula compute mean, standard deviation and 0, 025 and 0, 975-quantiles for the Rayleigh distribution with estimated parameters σ_{MM} and σ_{ML} for each n estimated in the previous point, as well as for true value of the parameter. Present the values in the table and comment on them.

Solution 2.

1. For $\sigma, y > 0$ the CDF F of the Rayleigh-distribution is given by

$$\begin{aligned} F(y) &= \int_0^y \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} dx \quad \left| \begin{array}{l} v := \frac{x^2}{2\sigma^2}, \quad dx = \frac{\sigma^2}{x} dv \end{array} \right. \\ &= \int_0^{\frac{y^2}{2\sigma^2}} e^{-v} dv \\ &= 1 - e^{-\frac{y^2}{2\sigma^2}} \end{aligned}$$

Then, for $u \in [0, 1]$, we have

$$\begin{aligned} u = F(y) &\iff u = 1 - e^{-\frac{y^2}{2\sigma^2}} \\ &\iff y = \sqrt{-2\sigma^2 \ln(1 - u)}. \end{aligned}$$

Thus,

$$F^{-1}(u) = \sqrt{-2\sigma^2 \ln(1 - u)}.$$

2. For X Rayleigh-distributed, we have

$$\mathbb{E}[X] = \sigma \sqrt{\pi/2}.$$

Therefore, the moment estimator of σ is

$$\sigma_{MM} = \sqrt{\frac{2}{\pi}} \frac{1}{n} \sum_{i=1}^n Z_i.$$

3. Let $Z_i = z_i, i = 1, \dots, n$, then the log-likelihood function of the Rayleigh-distribution is given by

$$\begin{aligned} L(\sigma) &= \ln \left(\prod_{i=1}^n f(z_i) \right) \\ &= \sum_{i=1}^n \left(\ln(z_i) - \frac{z_i^2}{2\sigma^2} \right) - n \ln(\sigma^2). \end{aligned}$$

To derive the maximum likelihood estimator we set

$$\begin{aligned} L'(\hat{\sigma}) = 0 &\iff \frac{1}{\hat{\sigma}^3} \sum_{i=1}^n z_i^2 - \frac{2n}{\hat{\sigma}} = 0 \\ &\iff \hat{\sigma} = \sqrt{\frac{1}{2n} \sum_{i=1}^n z_i^2}. \end{aligned}$$

We have

$$\begin{aligned} L''(\hat{\sigma}) &= -3\frac{1}{\hat{\sigma}^4} \sum_{i=1}^n z_i^2 + \frac{2n}{\hat{\sigma}^2} \\ &= 4n^2(1-3) \left(\sum_{i=1}^n z_i^2 \right)^{-1} < 0. \end{aligned}$$

Hence

$$\sigma_{ML} = \hat{\sigma} = \sqrt{\frac{1}{2n} \sum_{i=1}^n z_i^2}$$

is the maximum likelihood estimator.

4. The script can be found in Appendix 2.2.
5. Figure 2 displays the table:

n	Moment estimation	Maximum Likelihood
10	2.404019	2.496286
100	2.622280	2.635414
1000	2.955811	2.992975
10000	2.998461	3.000266
1000000	2.996741	2.997432

Figure 2: Table of estimated scale parameters using both moment estimator and maximum likelihood for different sample sizes. Samples are drawn according to an actual scale parameter $\sigma = 3$.

We can see that both estimators are improving with increasing sample size n . From an error of order 10^{-1} for $n = 10$, the error shrinks to an order of 10^{-3} for the larger sample sizes. The maximum likelihood estimator generally performs slightly better.

6. The closed form formula of the quantile function is as follows

$$Q(p, \sigma) = \sigma \sqrt{-2 \ln(1-p)}$$

for $\sigma > 0$, $p \in [0, 1)$. Here are computed means, standard deviations, 0, 025 and 0, 975-quantiles for the estimated parameters from Figure 2:

n	Moment Estimation				Maximum Likelihood			
	Mean	St.dev	0,025 quantile	0,975 quantile	Mean	St.dev	0,025 quantile	0,975 quantile
10	3.0130	1.5750	0.5410	6.5298	3.1286	1.6354	0.5617	6.7804
100	3.2865	1.7180	0.5901	7.1226	3.3030	1.7266	0.5930	7.1583
1000	3.7046	1.9365	0.6651	8.0286	3.7511	1.9608	0.6735	8.1295
10000	3.7580	1.9644	0.6747	8.1444	3.7603	1.9656	0.6751	8.1493
1000000	3.7559	1.9633	0.6743	8.1398	3.7567	1.9637	0.6745	8.1416

Figure 3: Computed means, standard deviations, 0, 025 and 0, 975-quantiles for the Rayleigh distribution corresponding to the estimated scale parameters from Figure 2.

And here are the same properties computed for the true parameter $\sigma = 3$:

True parameter $\sigma = 3$			
Mean	St.dev	0,025 quantile	0,975 quantile
3.7599	1.9654	0.6751	8.1486

Figure 4: Computed means, standard deviations, 0, 025 and 0, 975-quantiles for the Rayleigh distribution corresponding to the scale parameter $\sigma = 3$.

Problem 3. We assume the compound Poisson model for $J = 1000$ policies. In this model, the claim number from all policies is modelled as a random variable $N \sim Poiss(\lambda)$, ($\lambda = J\mu, \mu = 0.1$) and the claim size Z_i is distributed according to Weibull distribution with parameters $\alpha = 2, \beta = 3$. Then the sum of all claims can be defined as $X = \sum_{i=1}^N Z_i$.

1. Compute the expectation and the standard deviation of X using the closed-form formula. Calculate the pure premium for one policy and compare it with the expected value of X . What is the relation?
2. Write a program which simulates the sum of claims X and simulate $n = 10000$ simulations. Compute the mean and standard deviation from the simulation and compare it with point 1.
3. Assume excess of loss reinsurance, where the reinsurer covers the loss between 2.5 and 5. Write a program which would simulate reinsurance recoverables as well as sum of net claims in this case.
4. Calculate the pure reinsurance premium in this case as well as mean of the sum of net claims.

Solution 3.

1. Using the closed form formula, we have the expectation and variance of the Weibull distributed Z_1 :

$$\begin{aligned}\mathbb{E}[Z_1] &= \beta \cdot \Gamma\left(1 + \frac{1}{\alpha}\right) = 3 \cdot \Gamma\left(\frac{3}{2}\right) = \frac{3\sqrt{\pi}}{2} \\ \text{Var}(Z_1) &= \beta^2 \left[\Gamma\left(1 + \frac{2}{\alpha}\right) - \left(\Gamma\left(1 + \frac{1}{\alpha}\right)\right)^2 \right] = 9 \left(1 - \frac{\pi}{4}\right)\end{aligned}$$

and the Poisson distributed N :

$$\mathbb{E}[N] = J\mu = \text{Var}(N).$$

The expectation of X is then

$$\begin{aligned}\mathbb{E}[X] &= \mathbb{E}\left[\sum_{i=1}^N Z_i\right] = \mathbb{E}\left[\sum_{i=1}^{\infty} Z_i \mathbb{1}_{\{N \geq i\}}\right] \\ &= \mathbb{E}\left[\sum_{i=1}^{\infty} Z_i \sum_{j=i}^{\infty} \mathbb{1}_{\{N=j\}}\right] = \mathbb{E}\left[\sum_{j=1}^{\infty} \mathbb{1}_{\{N=j\}} \sum_{i=1}^j Z_i\right] \\ &= \sum_{j=1}^{\infty} \mathbb{E}\left[\mathbb{1}_{\{N=j\}} \sum_{i=1}^j Z_i\right] = \sum_{j=1}^{\infty} \mathbb{P}(N=j) \sum_{i=1}^j \mathbb{E}[Z_i] \\ &= \sum_{j=1}^{\infty} \mathbb{P}(N=j) \cdot j \cdot \mathbb{E}[Z_1] = \mathbb{E}[N] \cdot \mathbb{E}[Z_1] = \frac{3\sqrt{\pi}}{2} J\mu,\end{aligned}$$

using that Z_1 and N are independent and the Z_i are iid. We have

$$\begin{aligned}
\text{Var}(X) &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \\
&= \mathbb{E} \left[\sum_{k=1}^{\infty} \mathbb{1}_{\{N=k\}} \sum_{i=1}^k \left(Z_i^2 + 2 \sum_{j<i} Z_i Z_j \right) \right] - \frac{9\pi}{4} J^2 \mu^2 \\
&= \sum_{k=1}^{\infty} \mathbb{P}(N=k) \sum_{i=1}^k \left(\mathbb{E}[Z_i^2] + 2 \sum_{j<i} \mathbb{E}[Z_i] \mathbb{E}[Z_j] \right) - \frac{9\pi}{4} J^2 \mu^2 \\
&= \sum_{k=1}^{\infty} \mathbb{P}(N=k) \cdot \left(9k + \sum_{i=1}^k 2(i-1) \frac{9\pi}{4} \right) - \frac{9\pi}{4} J^2 \mu^2 \\
&= \sum_{k=1}^{\infty} \mathbb{P}(N=k) \cdot \left(9k + \frac{9\pi}{4} k(k-1) \right) - \frac{9\pi}{4} J^2 \mu^2 \\
&= 9\mathbb{E}[N] + \frac{9\pi}{4} (\mathbb{E}[N^2] - \mathbb{E}[N]) - \frac{9\pi}{4} J^2 \mu^2 \\
&= 9J\mu,
\end{aligned}$$

so $\text{sd}(X) = 3\sqrt{J\mu}$. The pure premium η is

$$\eta = \mu \mathbb{E}[Z_i] = \mu \frac{3\sqrt{\pi}}{2},$$

which is simply the expected value of X divided by the number of policies.

2. The program can be found in Appendix 2.3. Our calculated mean was $\approx 266,15$ and standard deviation $\approx 30,04$. These are very close to the theoretical mean $\frac{3\sqrt{\pi}}{2} J\mu \approx 265,87$ and standard deviation $3\sqrt{J\mu} = 30$ from point 1.
3. The function can be found in Appendix 2.4. After 10000 simulations it returned a mean of $\approx 58,56$ for the sum reinsurance recoverables and a mean $\approx 207,42$ for the sum of net claims.
4. For a claim Z and bounds a and b , the reinsurance recoverable is given by

$$Z^{re} = H(Z) = (Z - a)\mathbb{1}_{\{a \leq Z < b\}} + b\mathbb{1}_{\{Z \geq b\}}.$$

Then the expected reinsurance payoff is

$$\begin{aligned}
\mathbb{E}[Z^{re}] &= \mathbb{E}[H(Z)] = \mathbb{E}[(Z - a)\mathbb{1}_{\{a \leq Z < a+b\}} + b\mathbb{1}_{\{Z \geq a+b\}}] \\
&= \int_a^{a+b} (z - a)f(z)dz + \int_{a+b}^{\infty} bf(z)dz \\
&= \int_a^{a+b} (1 - F(z))dz
\end{aligned}$$

for pdf f and cdf F , using integration by parts. For our Weibull distribution this equals

$$\begin{aligned}
\int_a^{a+b} (1 - F(z))dz &= \int_a^{a+b} e^{-(\frac{z}{3})^2} dz = 3\sqrt{\pi} \left(\Phi \left(\frac{\sqrt{2}}{3}(a+b) \right) - \Phi \left(\frac{\sqrt{2}}{3}a \right) \right) \\
&= 3\sqrt{\pi} \left(\Phi \left(\frac{5\sqrt{2}}{3} \right) - \Phi \left(\frac{5\sqrt{2}}{6} \right) \right),
\end{aligned}$$

where Φ denotes the cdf of the standard normal distribution. So the pure reinsurance premium η^{re} for a single event is then

$$\eta^{re} = \mu \mathbb{E}[Z^{re}] = \mu 3\sqrt{\pi} \left(\Phi \left(\frac{5\sqrt{2}}{3} \right) - \Phi \left(\frac{5\sqrt{2}}{6} \right) \right) \approx 0,5854.$$

As you can see, this is approximately $\frac{1}{1000}$ th of what our simulations in 3. returned for the mean sum of reinsurance recoverables. As we simulated a portfolio of 1000 policies, these numbers match.

For the expected net claims, we can use the fact that Z^{re} is independent of N , as well. The same way as in 1., we obtain

$$\begin{aligned} \mathbb{E}[X - X^{re}] &= \mathbb{E}[N](\mathbb{E}[Z_1] - \mathbb{E}[Z_1^{re}]) \\ &= J\mu \left(\frac{3\sqrt{\pi}}{2} - 3\sqrt{\pi} \left(\Phi \left(\frac{5\sqrt{2}}{3} \right) - \Phi \left(\frac{5\sqrt{2}}{6} \right) \right) \right) \\ &\approx 207,33, \end{aligned}$$

which matches our simulated sum from 3.

2 Appendix

2.1 Problem 1.3 - Bond value function

```
def v_0(r, T, c, P):
    """
    Calculate the present value (price) of a fixed coupon bond.

    Parameters
    -----
    r : float
        The annual interest rate (yield to maturity), expressed as a decimal.
        Example: 0.05 for 5%.
    T : int or float
        The number of years until maturity.
    c : float
        The coupon rate per period, expressed as a decimal.
        Example: 0.06 for a 6% annual coupon.
    P : float
        The bond's principal or face value.

    Returns
    -----
    float
        The present value (price) of the bond.
    """

    # Start with the present value of the principal repayment at maturity:
    # P discounted back T periods at rate r.
```

```

sum = P * np.power(1 + r, -float(T))

# Initialize a counter for coupon payments
i = 1.0

# Loop over each coupon payment period (1 through T)
while i <= T:
    # Add the present value of each coupon payment (c * P) discounted i periods
    sum += c * P * np.power(1 + r, -float(i))
    i += 1.0

# Return the total present value (price)
return sum

```

2.2 Problem 2.4 - ML Estimator

```

def ray_ml(sample):
    """
    Compute the Maximum Likelihood Estimator (MLE) of the scale parameter (sigma)
    for a Rayleigh-distributed sample.

    Parameters
    -----
    sample : array-like
        A list or NumPy array of observed data points assumed to come from
        a Rayleigh distribution.

    Returns
    -----
    float
        The MLE estimate of the scale parameter sigma.
    """

    # Step 1: Compute the sum of squared observations.
    # For the Rayleigh distribution, each observation x_i enters as x_i^2.
    s = np.sum([x**2 for x in sample])

    # Step 2: Apply the MLE formula for sigma:
    # sigma_hat = sqrt( (1 / (2n)) * sum(x_i^2) )
    # where n is the number of observations.
    s = np.sqrt(s / (2 * len(sample)))

    # Step 3: Return the estimated scale parameter.
    return s

```

2.3 Problem 3.2 - Modeling claims

```
def sim_claims(n):
    """
    Simulate total claim amounts for a portfolio of 1000 policies where:
    - The number of claims follows a Poisson process with mean 100.
    - Individual claim sizes follow a Weibull(2, 3) distribution.

    Parameters
    -----
    n : int
        Number of simulations to run.

    Returns
    -----
    mean : float
        Estimated mean of the total claim amounts over n simulations.
    stdev : float
        Estimated standard deviation of the total claim amounts.
    """

    sims = np.zeros(n) # Store simulated total losses

    for i in range(n):
        # --- Step 1: Simulate the number of claims N ---
        # The loop below uses Algorithm 2.14 from Bølviken's book to
        # generate a Poisson-distributed count.
        y = 0
        N = 0
        while y < 100:
            u = np.random.uniform(0, 1)
            y = y - np.log(u)
            N += 1
        # At the end, (N - 1) is approximately Poisson(100).

        # --- Step 2: Simulate total claim severity ---
        # We utilize Algorithm 2.12 from Bølviken's book to generate
        # a Weibull(2, 3)-distributed claim size.
        x = 0
        for j in range(N - 1): # Each claim
            u = np.random.uniform(0, 1)
            # Weibull(2, 3): z = scale * (-ln(U))^(1/shape)
            z = 3 * np.power(-np.log(u), 0.5)
            x += z # Add claim size

        sims[i] = x # Store total claim amount for this simulation

    # --- Step 3: Compute statistics over all simulations ---
```

```

mean = np.mean(sims)
stdev = np.std(sims)

return mean, stdev

```

2.4 Problem 3.3 - Modeling Reinsurance Recoverables

```

def sim_reinsurance(n):
    """
    Simulate total reinsurance recoverables and net claims under an
    excess-of-loss reinsurance treaty.

    The reinsurer covers individual claim amounts in the layer (2.5, 5]:
        - The insurer pays all losses up to 2.5.
        - The reinsurer pays the part between 2.5 and 5.
        - Any amount above 5 remains with the insurer.

    The claim count process and claim size distribution are the same
    as in sim_claims().

    Parameters
    -----
    n : int
        Number of simulations to run.

    Returns
    -----
    mean_reinsurer : float
        Estimated mean of total reinsurance recoveries.
    mean_net : float
        Estimated mean of total net (insurer) claims.
    """

    sims = np.zeros((n, 2)) # Column 0: reinsurer, Column 1: net (insurer)

    for i in range(n):
        # --- Step 1: Simulate number of claims N ~ Poisson(100) ---
        y = 0
        N = 0
        while y < 100:
            u = np.random.uniform(0, 1)
            y = y - np.log(u)
            N += 1

        # --- Step 2: Simulate claim severities and apply reinsurance ---
        x = 0 # total (gross) claims
        x_re = 0 # total reinsurer recoverables

        for j in range(N - 1):

```

```
u = np.random.uniform(0, 1)
z = 3 * np.power(-np.log(u), 0.5) # Weibull(2,3)

# Reinsurance layer (2.5, 5]:
if z >= 2.5:
    # Reinsurer pays the part of the loss between 2.5 and 5
    x_re += np.min([z, 5]) - 2.5

x += z

# --- Step 3: Calculate totals for this simulation ---
net_claims = x - x_re          # insurer net = total - reinsured
sims[i, :] = [x_re, net_claims]

# --- Step 4: Return expected (mean) recoverables and net losses ---
return np.mean(sims[:, 0]), np.mean(sims[:, 1])
```