AM 3038 / AM 3084/ FM 3036

Optimal Insurance models

Report 3

Lasal Sandeepa Hettiarachchi (S14815)

Sahashra Dinasiri (S14980)

Udani Anupama (S15048)

Department of Mathematics
University of Colombo
Sri Lanka

January 11 2023

Table of Contents

Problem Description

The problem is given regarding the options of insurances available to 10 farmers who live in the same area and having the same size of paddy fields. The initial wealth level of them is 0.5 and they can earn 1 from farming during the good weather, but they can't earn anything during bad weather. The farmers are expected utility maximizers whose preferences over wealth are given by exponential utility functions with the coefficient of risk aversions $\gamma_i$of farmer i, for i=1, 2...,10.

γ_1=1, γ_2=1.5, γ_3=2, γ_4=2.5, γ_5=3, γ_6=3.5, γ_7=4, γ_8=4.5, γ_9=5, γ_{10}=5.5.

The payout of the full cover insurance is 1 and the payout of a partial cover insurance is equal to the corresponding coinsurance level. An insurance company is willing to sell a traditional insurance to cover the future agriculture risk of the farmers in area XYZ. By using the data from the weather station, the insurance company developed a rainfall index. According to that rainfall index, the coming year will have a good level of rainfall for farming with 2/3 chance. According to a pilot project the insurance company realized that their rainfall index is not perfectly correlated with the actual loss of the farmer. The full cover index insurance will pay 1 and the payout of a partial cover insurance is equal to the corresponding coinsurance level.

The following possibilities are available for each farmer.

1.Do not purchase insurance

2.Purchase traditional insurance with any coinsurance level.

3.Purchase index insurance with any coinsurance level.

Assumptions

It is assumed that the utility function is taken as,

$$u(wi)\begin{cases} \log(w_i)\ if\ gamma = 1 \\ \log\,(w_i{}^{1-gamma}\ /\ 1 - gamma) \end{cases}$$

Model Development

Gain of the farmers and the probability table

Gain	Probability
0	1/3
1	2/3

$E[P] = 0 \times 1/3 + 1 \times 2/3\ = 2/3$

	Good rainfall	Bad rainfall	
Good for farmers	5/9	1/9	2/3
Bad for farmers	1/9	2/9	1/3

	2/3	1/3	1

Task 1

$U_i(w_i)$ = utility function for farmer I
w_i = wealth level

The utility function relevant to the model is of the form:
$U_i(w_i) = \ln(w_i)$ for $\gamma_i = 1$
$U_i(w_i) = \ln(w_i$^$(1-\gamma_i))/(1-\gamma_i)$ for $\gamma_i > 1$

The expected utility for each farmer:
$E[U_i]$ = P (good year) * $U_i(w_i + 1)$ + P (bad year) * $U_i(w_i)$

p_i be the probability of a bad year for farmer i (since the probability of a good year is 2/3 and we know the index is not perfect, P (good year) = 1/9)

The expected utility for each farmer can be calculated using the above equation.

For the traditional insurance, let x_i be the coinsurance level for farmer i. Then, the expected utility for each farmer can be calculated as:
E [U_i (traditional insurance)] = P (good year) * $U_i(w_i + 1 - x_i)$ + P (bad year) * $U_i(w_i + 1)$

For the index insurance, let y_i be the coinsurance level for farmer i. Then, the expected utility for each farmer can be calculated as:
E [U_i (index insurance)] = P (index indicates bad year) * P (bad year index indicates bad year) * $U_i(w_i + 1 - y_i)$ + P (index indicates good year) * P (good year index indicates good year) * $U_i(w_i + 1)$

Given the expected utility for each option and the actuarial fair premium for each insurance, the best choice for each farmer can be determined by finding the option with the highest expected utility.

To solve this problem, “expected utility” is needed to calculate for each option for each farmer, given their coefficient of risk aversion (γ_i)Then the expected utility can be obtained for each option to determine the best choice for each farmer.

It's worth noting that in reality this would require to factor in the cost of the insurance in the final decision of the farmers. However, since it's not specified in the problem statement and it's the main driving factor for decision making, it's not included in the mathematical model.

Task 2

To extend the problem as described, the expected utility of each farmer can be calculated by considering the different options available to them. The expected utility of a given choice can be calculated using the following formula:

$Eu = \Sigma(p * u(w))$

where:

Eu = expected utility
p = probability of the outcome
u(w) = utility of wealth (w)

For a given farmer, the options available to them are:

Do not purchase insurance:
In this case, the expected utility is equal to the expected wealth, which is:
Eu = 0.5 * u(0.5) + (2/3) * u(1.5) + (1/3) * u(0.5)
Purchase traditional insurance with any coinsurance level c:
In this case, the expected utility is equal to:
Eu = (2/3) * u(1-c+0.5) + (1/3) * u(1-c+0.5)
Purchase index insurance with any coinsurance level c:
In this case, the expected utility is equal to:
Eu = (8/9) * u(1-c+0.5) + (1/9) * u(0.5)
Here u(w) is the utility function with coefficient of risk aversion γi of farmer i , u(w) = ln(w) if γ_i =1, u(w) = w^(1-γ_i) / (1-γ_i) if $\gamma_i \neq 1$

Then the farmer will choose the option with the highest expected utility. It is difficult to provide a general solution for which option will be the best choice for each farmer, as it depends on the individual utility function and the corresponding coefficient of risk aversion γi for each farmer. However, for each of the farmers you can calculate the expected utility for each of the three options, and then compare them. Whichever option has the highest expected utility for that farmer, that should be the best choice for them.

Note that, the premium is 50% more than the actuarial fair premium, so the premium for both the traditional and index insurance will be 1.5 times the actuarial fair premium.

Path to the Solution

Task 1

```
import numpy as np
from scipy.optimize import minimize_scalar


def utility(wi, gamma):
    if gamma == 1:
        return np.log(wi)
    else:
        return (np.log(wi**(1-gamma))/(1-gamma))


# Define the coefficients of risk aversion for each farmer
gammas = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5]
p = 1/9
```

```
# For each farmer, calculate the expected utility for each option
for i, gamma in enumerate(gammas):
    wi = 0.5
    eu_no_ins = p*utility(wi, gamma) + (1-p)*utility(wi+1, gamma)
    res = minimize_scalar(lambda x: -(p*utility(wi+1-x, gamma) + (1-p)*utility(wi+1, gamma)), bounds=(0,1),
method='bounded')
    eu_traditional_ins = -res.fun
    res = minimize_scalar(lambda y: -((1/9)*p*utility(wi+1-y, gamma) + (8/9)*(1-p)*utility(wi+1, gamma)),
bounds=(0,1), method='bounded')
    eu_index_ins = -res.fun

    # Compare the expected utility for each option to determine the best choice
    best_choice = max(eu_no_ins, eu_traditional_ins, eu_index_ins)
    if best_choice == eu_no_ins:
        print("Farmer {}: Best choice is to not purchase insurance, with expected utility {}".format(i+1, eu_no_ins))
    elif best_choice == eu_traditional_ins:
        print("Farmer {}: Best choice is to purchase traditional insurance, with expected utility {}".format(i+1,
eu_traditional_ins))
    else:
        print("Farmer {}: Best choice is to purchase index insurance, with expected utility {}".format(i+1,
eu_index_ins))
```

```
"/Users/lasalhettiarachchi/Development/Computational Modeling/venv/bin/python" /Users/lasalhettiarachchi/Development/Computational Modeling/OptimalInsuaranceModel.py
Farmer 1: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 2: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 3: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 4: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 5: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 6: Best choice is to purchase traditional insurance, with expected utility 0.40546466656202873
Farmer 7: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 8: Best choice is to purchase traditional insurance, with expected utility 0.40546466656202873
Farmer 9: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288
Farmer 10: Best choice is to purchase traditional insurance, with expected utility 0.4054646665620288

Process finished with exit code 0
```

Task 2

```
import math

# Define the utility function for a given wealth and coefficient of risk aversion
def utility(wealth, gamma):
    if gamma == 1:
        return math.log(wealth)
    else:
        return (wealth ** (1 - gamma)) / (1 - gamma)
```

```python
# Define the expected utility for each option for a given farmer
def expected_utility(gamma, option, c):
    if option == 1:
        return 0.5 * utility(0.5, gamma) + (2/3) * utility(1.5, gamma) + (1/3) * utility(0.5, gamma)
    elif option == 2:
        return (2/3) * utility(1 - c + 0.5, gamma) + (1/3) * utility(1 - c + 0.5, gamma)
    elif option == 3:
        return (8/9) * utility(1 - c + 0.5, gamma) + (1/9) * utility(0.5, gamma)
    else:
        return None

# List of coefficients of risk aversion for each farmer
gammas = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5]

# For each farmer, calculate the expected utility for each option
for i, gamma in enumerate(gammas):
    print(f"Farmer {i+1} (gamma = {gamma})")
    for option in [1, 2, 3]:
        for c in [0, 0.1, 0.2, 0.3, 0.4, 0.5]:
            eu = expected_utility(gamma, option, c)
            print(f"  Option {option} (c = {c}): {eu}")
    print()
```

```
Farmer 1 (gamma = 1)
  Option 1 (c = 0): -0.30731257839451154
  Option 1 (c = 0.1): -0.30731257839451154
  Option 1 (c = 0.2): -0.30731257839451154
  Option 1 (c = 0.3): -0.30731257839451154
  Option 1 (c = 0.4): -0.30731257839451154
  Option 1 (c = 0.5): -0.30731257839451154
  Option 2 (c = 0): 0.4054651081081633
  Option 2 (c = 0.1): 0.33647223662121284
  Option 2 (c = 0.2): 0.26236426446749106
  Option 2 (c = 0.3): 0.1823215567939546
  Option 2 (c = 0.4): 0.09531017980432493
  Option 2 (c = 0.5): 0.0
  Option 3 (c = 0): 0.28339707603393
  Option 3 (c = 0.1): 0.22207007915663973
  Option 3 (c = 0.2): 0.15619632613110923
  Option 3 (c = 0.3): 0.08504725264352127
  Option 3 (c = 0.4): 0.007703806430517129
  Option 3 (c = 0.5): -0.07701635339554948
```

```
Farmer 2 (gamma = 1.5)
  Option 1 (c = 0): -3.445684711858793
  Option 1 (c = 0.1): -3.445684711858793
  Option 1 (c = 0.2): -3.445684711858793
  Option 1 (c = 0.3): -3.445684711858793
  Option 1 (c = 0.4): -3.445684711858793
  Option 1 (c = 0.5): -3.445684711858793
  Option 2 (c = 0): -1.632993161855452
  Option 2 (c = 0.1): -1.690308509457033
  Option 2 (c = 0.2): -1.7541160386140582
  Option 2 (c = 0.3): -1.8257418583505538
  Option 2 (c = 0.4): -1.9069251784911845
  Option 2 (c = 0.5): -2.0
  Option 3 (c = 0): -1.7658191577322007
  Option 3 (c = 0.1): -1.8167661333780505
  Option 3 (c = 0.2): -1.873483937073184
  Option 3 (c = 0.3): -1.9371513323945133
  Option 3 (c = 0.4): -2.0093142836306295
  Option 3 (c = 0.5): -2.092047458305132
```

```
Farmer 3 (gamma = 2)
  Option 1 (c = 0): -2.111111111111111
  Option 1 (c = 0.1): -2.111111111111111
  Option 1 (c = 0.2): -2.111111111111111
  Option 1 (c = 0.3): -2.111111111111111
  Option 1 (c = 0.4): -2.111111111111111
  Option 1 (c = 0.5): -2.111111111111111
  Option 2 (c = 0): -0.6666666666666666
  Option 2 (c = 0.1): -0.7142857142857142
  Option 2 (c = 0.2): -0.7692307692307692
  Option 2 (c = 0.3): -0.8333333333333334
  Option 2 (c = 0.4): -0.909090909090909
  Option 2 (c = 0.5): -1.0
  Option 3 (c = 0): -0.8148148148148148
  Option 3 (c = 0.1): -0.8571428571428571
  Option 3 (c = 0.2): -0.9059829059829059
  Option 3 (c = 0.3): -0.9629629629629629
  Option 3 (c = 0.4): -1.0303030303030303
  Option 3 (c = 0.5): -1.1111111111111112
```

```
Farmer 4 (gamma = 2.5)
  Option 1 (c = 0): -1.8132733155042469
  Option 1 (c = 0.1): -1.8132733155042469
  Option 1 (c = 0.2): -1.8132733155042469
  Option 1 (c = 0.3): -1.8132733155042469
  Option 1 (c = 0.4): -1.8132733155042469
  Option 1 (c = 0.5): -1.8132733155042469
  Option 2 (c = 0): -0.36288736930121157
  Option 2 (c = 0.1): -0.4024544070135794
  Option 2 (c = 0.2): -0.4497733432343739
  Option 2 (c = 0.3): -0.5071505162084872
  Option 2 (c = 0.4): -0.5778561146942983
  Option 2 (c = 0.5): -0.6666666666666666
  Option 3 (c = 0): -0.5320796708415354
  Option 3 (c = 0.1): -0.5672503710303068
  Option 3 (c = 0.2): -0.6093116476710132
  Option 3 (c = 0.3): -0.6603135792035582
  Option 3 (c = 0.4): -0.7231630000798348
  Option 3 (c = 0.5): -0.8021057129441622
```

```
Farmer 5 (gamma = 3)
  Option 1 (c = 0): -1.8148148148148149
  Option 1 (c = 0.1): -1.8148148148148149
  Option 1 (c = 0.2): -1.8148148148148149
  Option 1 (c = 0.3): -1.8148148148148149
  Option 1 (c = 0.4): -1.8148148148148149
  Option 1 (c = 0.5): -1.8148148148148149
  Option 2 (c = 0): -0.2222222222222222
  Option 2 (c = 0.1): -0.25510204081632654
  Option 2 (c = 0.2): -0.29585798816568043
  Option 2 (c = 0.3): -0.34722222222222227
  Option 2 (c = 0.4): -0.4132231404958677
  Option 2 (c = 0.5): -0.5
  Option 3 (c = 0): -0.41975308641975306
  Option 3 (c = 0.1): -0.44897959183673464
  Option 3 (c = 0.2): -0.4852071005917159
  Option 3 (c = 0.3): -0.5308641975308642
  Option 3 (c = 0.4): -0.5895316804407713
  Option 3 (c = 0.5): -0.6666666666666666
```

```
Farmer 6 (gamma = 3.5)
  Option 1 (c = 0): -1.9823880483111165
  Option 1 (c = 0.1): -1.9823880483111165
  Option 1 (c = 0.2): -1.9823880483111165
  Option 1 (c = 0.3): -1.9823880483111165
  Option 1 (c = 0.4): -1.9823880483111165
  Option 1 (c = 0.5): -1.9823880483111165
  Option 2 (c = 0): -0.1451549477204846
  Option 2 (c = 0.1): -0.17248046014867688
  Option 2 (c = 0.2): -0.20758769687740333
  Option 2 (c = 0.3): -0.2535752581042436
  Option 2 (c = 0.4): -0.31519424437870813
  Option 2 (c = 0.5): -0.4
  Option 3 (c = 0): -0.38044236461786984
  Option 3 (c = 0.1): -0.4047317089984852
  Option 3 (c = 0.2): -0.43593814164624206
  Option 3 (c = 0.3): -0.47681597384787777
  Option 3 (c = 0.4): -0.5315884060918463
  Option 3 (c = 0.5): -0.606971299977439
```

```
Farmer 7 (gamma = 4)
  Option 1 (c = 0): -2.288065843621399
  Option 1 (c = 0.1): -2.288065843621399
  Option 1 (c = 0.2): -2.288065843621399
  Option 1 (c = 0.3): -2.288065843621399
  Option 1 (c = 0.4): -2.288065843621399
  Option 1 (c = 0.5): -2.288065843621399
  Option 2 (c = 0): -0.09876543209876543
  Option 2 (c = 0.1): -0.12147716229348884
  Option 2 (c = 0.2): -0.15172204521316945
  Option 2 (c = 0.3): -0.19290123456790123
  Option 2 (c = 0.4): -0.2504382669671925
  Option 2 (c = 0.5): -0.3333333333333333
  Option 3 (c = 0): -0.3840877914951989
  Option 3 (c = 0.1): -0.4042759961127308
  Option 3 (c = 0.2): -0.4311603364857802
  Option 3 (c = 0.3): -0.4677640603566529
  Option 3 (c = 0.4): -0.5189080891560229
  Option 3 (c = 0.5): -0.5925925925925926
```

```
Farmer 8 (gamma = 4.5)
  Option 1 (c = 0): -2.739821054590176
  Option 1 (c = 0.1): -2.739821054590176
  Option 1 (c = 0.2): -2.739821054590176
  Option 1 (c = 0.3): -2.739821054590176
  Option 1 (c = 0.4): -2.739821054590176
  Option 1 (c = 0.5): -2.739821054590176
  Option 2 (c = 0): -0.06912140367642125
  Option 2 (c = 0.1): -0.0880002347697331
  Option 2 (c = 0.2): -0.11405917410846336
  Option 2 (c = 0.3): -0.15093765363347833
  Option 2 (c = 0.4): -0.20467158725890133
  Option 2 (c = 0.5): -0.2857142857142857
  Option 3 (c = 0): -0.4206065968864938
  Option 3 (c = 0.1): -0.43738778008054885
  Option 3 (c = 0.2): -0.4605512817149757
  Option 3 (c = 0.3): -0.49333215240387795
  Option 3 (c = 0.4): -0.5410956489598095
  Option 3 (c = 0.5): -0.6131336031423733
```

```
Farmer 9 (gamma = 5)
  Option 1 (c = 0): -3.3662551440329214
  Option 1 (c = 0.1): -3.3662551440329214
  Option 1 (c = 0.2): -3.3662551440329214
  Option 1 (c = 0.3): -3.3662551440329214
  Option 1 (c = 0.4): -3.3662551440329214
  Option 1 (c = 0.5): -3.3662551440329214
  Option 2 (c = 0): -0.04938271604938271
  Option 2 (c = 0.1): -0.06507705122865473
  Option 2 (c = 0.2): -0.08753194916144391
  Option 2 (c = 0.3): -0.12056327160493828
  Option 2 (c = 0.4): -0.17075336384126763
  Option 2 (c = 0.5): -0.25
  Option 3 (c = 0): -0.4883401920438957
  Option 3 (c = 0.1): -0.5022907122032486
  Option 3 (c = 0.2): -0.522250621476839
  Option 3 (c = 0.3): -0.5516117969821673
  Option 3 (c = 0.4): -0.5962252123033489
  Option 3 (c = 0.5): -0.6666666666666666
```

```
Farmer 10 (gamma = 5.5)
  Option 1 (c = 0): -4.214156225586205
  Option 1 (c = 0.1): -4.214156225586205
  Option 1 (c = 0.2): -4.214156225586205
  Option 1 (c = 0.3): -4.214156225586205
  Option 1 (c = 0.4): -4.214156225586205
  Option 1 (c = 0.5): -4.214156225586205
  Option 2 (c = 0): -0.03584072783221842
  Option 2 (c = 0.1): -0.048889019316518394
  Option 2 (c = 0.2): -0.06824053151788406
  Option 2 (c = 0.3): -0.0978299606883656
  Option 2 (c = 0.4): -0.14471728392043529
  Option 2 (c = 0.5): -0.2222222222222222
  Option 3 (c = 0): -0.5905600790106021
  Option 3 (c = 0.1): -0.6021585603299798
  Option 3 (c = 0.2): -0.6193599045089715
  Option 3 (c = 0.3): -0.6456616193271774
  Option 3 (c = 0.4): -0.6873392399779059
  Option 3 (c = 0.5): -0.7562325184683832
```

Post optimal analysis

$W \ = \ w - (1+\lambda)E(\propto L) - L + \propto L$
$W - terminal\ wealth$
$w - initial\ wealth$
$\propto -coinsuarance\ level$
$L - Loss$

At the actual fair premium price, $\lambda = 0$ then,

$W \ = \ w - \propto E(L) - L + \propto L$

In this problem, we consider a gain and not having a gain from the farms, therefore for an individual farmer the model can be defined as follows,

$W = w - \pi + C + G$
$W - terminal\ wealth$
$w - initial\ wealth$
$\pi - Premium\ value$
$C - Compensation\ (0 \ \leq C \ \ \leq 1)$
$G - Gain\ \ (0 \ \leq G \ \ \leq 1)$

$\pi = (1+\ \lambda)\ E\ (C)$
At actual fair premium price, $\lambda = 0$
$\pi = E\ (C)$
$C = \propto (1 - G)$
$E\ (C) = E\big(\propto (1 - G)\big)$
$E\ (C) = \propto E(1 - G)$
$\pi = \propto E(1 - G)$