SCC.363 Security and Risk Coursework 2

Academic Honesty and Integrity

Students at Lancaster University are part of an academic community that values trust, fairness and respect and actively encourages students to act with honesty and integrity. It is a University policy that students take responsibility for their work and comply with the university's standards and requirements- found in the Manual of Academic Regulations and Practice. By submitting their answers students will be confirming that the work submitted is completely their own. Academic misconduct regulations are in place for all forms of assessment and students may familiarise themselves with this via the university website:
https://www.lancaster.ac.uk/academic-standards-and-quality/regulations-policies-and-committees/manual-of-academic-regulations-and-procedures/

Plagiarism

Plagiarism involves the unacknowledged use of someone else's work and passing it off as if it were one's own. This covers every form of submitted work, from written essays, video vignettes, and coding exercises. However, deliberately plagiarism with the intent to deceive and gain academic benefit is unacceptable. This is a conscious, pre-meditated form of cheating and is regarded as a serious breach of the core values of the University. More information may be found via the plagiarism framework website. All coursework is to be submitted electronically and will be run through our plagiarism detection mechanisms. Please ensure you are familiar with the University's Plagiarism rules and if you are in any doubt please contact your module tutor.
https://www.lancaster.ac.uk/academic-standards-and-quality/regulations-policies-and-committees/principles-policies-and-guidelines/plagiarism-frame

This is an individual assignment that will count for 30% of your overall marks for this module. The marks will be based on the correctness of the solution.

Learning objectives

- Develop appreciation and understanding of security tools.
- Formulate troubleshooting methods to identify/solve problems.
- Evaluate information to critically argument solution choices.
- Effectively communicate ideas.

Submission requirements

Prepare and submit your coding solutions on Coderunner. For all coding solutions, you must use Python3. You can use modules from standard Python3. Your code should include appropriate comments explaining what you do and why.

BE AWARE OF THE FOLLOWING:
Multiple submissions on Coderunner to check your code's correctness against checked test cases and resubmission of your solution will result in a corresponding penalty per repeat submission (i.e., 0% for first re-submission, -5% for second re-submission, -10%, etc.)

Marking guidelines

Task 1. Weight 30% of total marks
- Marks will be allocated based on the correctness of your solution.

Task 2. Weight 30% of total marks
- Marks will be allocated based on the correctness of your solution.

Task 3. Weight 40% of total marks
- Marks will be allocated based on the correctness of your solution.

Deadline for submissions: Friday 17th March 16:00

Task 1 – Calculation of annualized loss expectancy

The annualized loss expectancy (ALE) is an important metric to quantify the exposure of an asset caused by certain risks. It is equal to the product of the annual rate of occurrence (ARO) and the single loss expectancy (SLE). The ARO is an estimate of how often a risk would be successful in exploiting a vulnerability. The SLE is the monetary value expected from the occurrence of a risk on an asset and it is equal to the product of the typical asset value (AV) and the exposure factor (EF), where the EF represents the impact of the risk over the asset or percentage of asset lost. In conclusion of the above, we have the following two relationships:

$$\mathrm{ALE} = \mathrm{ARO} \times \mathrm{SLE} \qquad (\mathrm{Eq.}\,1)$$

$$\mathrm{SLE} = \mathrm{AV} \times \mathrm{EF} \qquad (\mathrm{Eq.}\,2)$$

(1) Consider a specific cyber risk on an asset. Assume that the AV follows a triangular distribution with lower limit $\boldsymbol{a} = £10000$, upper limit $\boldsymbol{b} = £35000$, and mode $\boldsymbol{c} = £18000$.
   (i) Find the probability $\mathbf{prob1}$ that the AV is no greater than $\mathbf{point1} = £12000$.
   (ii) Find the probability $\mathbf{prob2}$ that the AV is greater than $\mathbf{point2} = £25000$.
   (iii) Find the mean $\mathbf{MEAN\_t}$ and median $\mathbf{MEDIAN\_t}$ of the AV.

(2) We have collected the numbers of annual occurrences of the cyber risk for 15 years, given by the data set $\mathbf{data} = (11, 15, 9, 5, 3, 14, 16, 15, 12, 10, 11, 4, 7, 12, 6)$. Calculate the mean $\mathbf{MEAN\_d}$ and variance $\mathbf{VARIANCE\_d}$ of the data set.

(3) The Monte Carlo method has been widely used in risk analysis to generate draws from a probability distribution. It relies on repeated random sampling to obtain numerical results. Suppose that two flaws, A and B, are considered in the above problem. Each flaw will cause certain amount of impact. In particular, the impact caused by flaw A follows the log-normal distribution with $\boldsymbol{\mu} = 0$ and $\boldsymbol{\sigma} = 3$. The impact caused by flaw B follows the Pareto distribution with $\boldsymbol{x_m} = 1$ and $\boldsymbol{\alpha} = 4$. The total impact is the sum of the impacts caused by the two flaws. Use Monte Carlo method to simulate the probability distribution of the total impact. Specifically:
   (i) Randomly sample $\mathbf{num} = 500000$ points for the total impact.
   (ii) Based on your sampling points, derive the probability $\mathbf{prob3}$ that the total impact is greater than $\mathbf{point3} = 30$.
   (iii) Based on your sampling points, derive the probability $\mathbf{prob4}$ that the total impact is between $\mathbf{point4} = 50$ and $\mathbf{point5} = 100$.

(4) Assume that the median of the triangular distribution $\mathbf{MEDIAN\_t}$ derived in (1)-(iii) is used as the AV in (Eq. 2), the mean of the data set $\mathbf{MEAN\_d}$ derived in (2) is used as the ARO in (Eq. 1), and the probability $\mathbf{prob3}$ derived in (3)-(ii) is used as the EF in (Eq. 2). Calculate the value of the **ALE**.

Please use the following code

```
from dhCheck_Task1 import dhCheckCorrectness
def Task1(a, b, c, point1, point2, data, mu, sigma, xm, alpha,
num, point3, point4, point5):
    # TODO

    return (prob1, prob2, MEAN_t, MEDIAN_t, MEAN_d, VARIANCE_d,
prob3, prob4, ALE)
```

Note: do not truncate the results!

Inputs format:

a, b, c, point1, point2, mu, sigma, xm, alpha, num, point3, point4, point5 = 10000, 35000, 18000, 12000, 25000, 0, 3, 1, 4, 500000, 30, 50, 100
data = [11, 15, 9, 5, 3, 14, 16, 15, 12, 10, 11, 4, 7, 12, 6]

Reading:
Annualized loss expectancy: https://en.wikipedia.org/wiki/Annualized_loss_expectancy
Single-loss expectancy: https://en.wikipedia.org/wiki/Single-loss_expectancy
Triangular distribution: https://en.wikipedia.org/wiki/Triangular_distribution
Log-normal distribution: https://en.wikipedia.org/wiki/Log-normal_distribution
Pareto distribution: https://en.wikipedia.org/wiki/Pareto_distribution
Monte Carlo method: https://en.wikipedia.org/wiki/Monte_Carlo_method

Task 2 – Probability theory

A breach management process has two phases, discovery (denoted by random variable X) and rectification (denoted by random variable Y). The breach management process is implemented on **num** = 120 cases and the joint distribution of the times required to complete the stages are provided by the following **table**. In particular, the second row lists the possible days for X, and the second column lists the possible days for Y. For each combination of the values of X and Y, the number at the intersection block is the frequency of this combination. For example, among the 120 cases, there are 6 cases that take X 5 days to complete and take Y 3 days to complete.

		X			
		5	6	7	8
Y	3	6	10	11	9
	4	9	12	15	8
	5	7	14	10	9

(1) Calculate the probability **prob1** of **eventA,** X = 7**,** and the probability **prob2** of **eventB**, 3 ≤ Y ≤ 4. Determine whether these two events are independent. Use a Boolean output **IsInd** to indicate the result, such that **IsInd** $= \mathbf{1}$ if they are independent, and **IsInd** $= \mathbf{0}$ if not.

(2) Another screening procedure was also tested on the 120 cases. Denote by T the event of being tested positive. The following probabilities, **probs**, were obtained: P(T|Y=3) = 0.7, P(T|Y=4) = 0.6, P(T|Y=5) = 0.5, P(T|X=5) = 0.63, P(T|X=6) = 0.44, and P(T|X=7) = 0.36. Find the probability **prob3** of being tested positive and the probability **prob4** of X=8 given that a case is tested positive.

Please use the following code

```
from dhCheck_Task2 import dhCheckCorrectness
def Task2(num, table, eventA, eventB, probs):
    # TODO

    return (prob1, prob2, IsInd, prob3, prob4)
```

Note: do not truncate the results!

Inputs format:

num = 120
eventA = 2 # column 2 of table
eventB = [0, 1] # row 0 and row 1 of table
probs = [0.7, 0.6, 0.5, 0.63, 0.44, 0.36]

table = [[6, 10, 11, 9], [9, 12, 15, 8], [7, 14, 10, 9]]

Reading:
Bayes' theorem: https://en.wikipedia.org/wiki/Bayes%27_theorem
Independence: https://en.wikipedia.org/wiki/Independence_(probability_theory)

Task 3 – Linear regression and linear programming

A company has been using five types of security controls to safeguard its computer network. To help determine a better deployment, the company would like to first understand the safeguard effects and maintenance loads of the security controls. Assume that both the total safeguard effect y and the total maintenance load z are linear functions of the numbers of the security control and can be expressed as $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5$ and $z = d_0 + d_1x_1 + d_2x_2 + d_3x_3 + d_4x_4 + d_5x_5$, respectively, where x_i is the number of the i-th type of security controls applied, b_i is the associated weight which represents the unit effect, and d_i is the associated weight which represents the unit load. The company has collected 10 historical pairs of joint input $\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5]$ and outputs $\boldsymbol{y}$ and $\boldsymbol{z}$, as listed in the following table.

$\boldsymbol{y}$	$\boldsymbol{z}$	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$	$\boldsymbol{x_5}$
176	352	5	3	8	9	4
170	384	4	7	3	3	9
215	471	8	7	6	9	6
146	358	8	2	7	3	6
228	412	2	2	9	10	10
145	345	5	5	10	4	3
183	449	5	10	6	2	8
151	357	7	4	2	3	8
160	366	8	6	2	7	4
151	349	8	3	3	5	6

(1) Based on the table, use linear regression to derive the underlying **weights_b** $(b_0, b_1, b_2, b_3, b_4, b_5)$ and **weights_d** $(d_0, d_1, d_2, d_3, d_4, d_5)$.

(2) Now assume that the company has already deployed the first four types of security controls with $\mathbf{num1} = x_1 = 5$, $\mathbf{num2} = x_2 = 6$, $\mathbf{num3} = x_3 = 8$ and $\mathbf{num4} = x_4 = 4$. It aims to achieve a total safeguard effect of $\mathbf{bound\_y} \geq 160$ and total maintenance load of $\mathbf{bound\_z} \leq 600$. Based on the weights $(b_0, b_1, b_2, b_3, b_4, b_5)$ and $(d_0, d_1, d_2, d_3, d_4, d_5)$ obtained in part (1), find the smallest value $\mathbf{s\_num5}$ and the largest value $\mathbf{l\_num5}$ of x_5 that can achieve this objective.

(3) The company now aims to strengthen its safeguard. The current deployment of the security controls is $x_1 = 3$, $x_2 = 5$, $x_3 = 4$, $x_4 = 2$ and $x_5 = 1$ (**x_initial**). The company plans to enhance the total safeguard effect to $y \geq 1000$ (**se_bound**). Meanwhile, the total maintenance load needs to be no greater than 2000 (**ml_bound**). In addition, there is an upper bound for the number of each type of security controls. Under these constraints, the company aims to minimize the total cost. The cost (**c**) and number bound (**x_bound**) for each type of security controls are given in the table below. The safeguard effect and the maintenance load are the ones derived in (1). Use linear programming to find how many *additional* security controls (**x_add**) for each type should be deployed. (Notice that the company already has a base deployment and wants to add more security controls to strengthen the safeguard.)

	1	2	3	4	5
Cost	11	6	8	10	9
Number bound	30	50	20	45	50

Please use the following code

```
from dhCheck_Task3 import dhCheckCorrectness
def Task3(x, y, z, num1, num2, num3, num4, bound_y, bound_z, c, se_bound, ml_bound, x_bound, x_initial):
    # TODO

    return (weights_b, weights_d, s_num5, l_num5, x_add)
```

Note:

(i) x_add must be the additional security controls, not the total security controls.

(ii) x_add should be of type array. You may have to use the dot operation to extract x_add from the solution returned by linprog.

(iii) You must round x_add to be proper integers (which must still satisfy all the constraints).

Inputs format:

x = [[5,4,8,8,2,5,5,7,8,8],[3,7,7,2,2,5,10,4,6,3],[8,3,6,7,9,10,6,2,2,3],[9,3,9,3,10,4,2,3,7,5],[4,9,6,6,10,3,8,8,4,6]]
y = [176,170,215,146,228,145,183,151,160,151]
z = [352,384,471,358,412,345,449,357,366,349]
num1, num2, num3, num4, bound_y, bound_z = 5, 6, 8, 4, 160, 600
c = [11, 6, 8, 10, 9]
se_bound = 1000
ml_bound = 2000
x_bound = [30,50,20,45,50]
x_initial = [3,5,4,2,1]

Reading:
Linear regression: https://en.wikipedia.org/wiki/Linear_regression
Linear programming: https://en.wikipedia.org/wiki/Linear_programming