This **make-up** assignment is due **September 6, 2024**. There will be no extensions for this assignment under any circumstances. You may work in teams of up to four people. If you work in teams, make sure you include everyone's name on the assignment and there is only one submission per team. Please submit one html or pdf file. Show all your work. Use as few Python packages as possible to complete the coding portion of this assignment. Email your solutions to **charles.tutoring@gmail.com**.

1. Consider the cash flows in the following table.

Time	0	1	2	3
Cash Flow	-100	60	20	50

(a) Write a Python function that gives the NPV of the cash flows for a given discount rate.

(b) Use part (a) to create a function that numerically computes the derivative of the NPV with respect to the discount rate.

(c) Newton's method is an algorithm to find the roots of a differentiable function f. The first step is to make an initial guess x_0. Then

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

for $n = 1, 2, \ldots$. Apply Newton's method to find the smallest positive root of the function you created in (a). Make sure your solution is accurate to at least two basis points. The solution is called the *internal rate of return* or *IRR*.

(d) Use `root_scalar` in `scipy.optimize` to check (c).

2. Use partition pairs of the form $P = (1, b^{1/n}, b^{2/n}, \ldots, b)$ and $T = (1, b^{1/n}, b^{2/n}, \ldots, b^{(n-1)/n})$ to analytically compute

$$\int_1^b \frac{dx}{x^p}$$

for $b,\ p > 1$.

x_i	1	2	3	4	5
y_i	0.87	3.37	2.37	6.79	9.05

3. Define an inner product on the vector space of functions from $\{1,2,3,4,5\}$ to $\mathbb{R}$ as

$$\langle f, g\rangle = \sum_{i=1}^{5} f(x_i)g(x_i).$$

(a) Orthogonalize the first two elements of the basis $(1, x, x^2, x^3, x^4)$ using the inner product defined above.

(b) Project the function f with the xy-coordinates in the table onto the two orthogonal basis elements you found in (a), and then rewrite your solution in the form $a + bx$. Due to the Projection Theorem, the function $g(x) = \alpha + \beta x$ that minimizes

$$\|f - g\|^2 = \sum_{i=1}^{5} [f(x_i) - g(x_i)]^2 = \sum_{i=1}^{5} [y_i - g(x_i)]^2$$

is your result. Note: This only holds on the domain $\{1,2,3,4,5\}$.

(c) Define

$$\mathcal{L}_2(\alpha, \beta) = \sum_{i=1}^{5} (y_i - \alpha - \beta x_i)^2 .$$

Minimize $\mathcal{L}_2$ with respect to α and β to analytically verify (b). Remember to check whether your solution is a minimum or a maximum.

(d) Use `scatter` to graph the points in the table. Also, use `plot` to graph the line corresponding to the coefficients you found in (b) on the same figure. Make sure to add a legend so that it is easy to interpret your results.

4. The central limit theorem tells us that if we independently sample $X_1, X_2, \ldots, X_n$ from an arbitrary distribution $\mathcal{D}$ with finite mean μ and variance σ^2, then $\frac{\sqrt{n}}{\sigma}(\bar{X} - \mu) \xrightarrow{d} \mathcal{N}(0, 1^2)$ as $n \to \infty$. For problem 4, we will assume $X_k \sim \mathcal{P}(1)$, so $\mu = 1$ and $\sigma^2 = 1$.

(a) Use `poisson.rvs` from `scipy.stats` to generate four arrays of size $(n, 100000)$ where n =10, 50, 100, and 1000. Set `mu = 1` in every case; this is the parameter λ from class. For each array, take the mean of each column via `array.mean(axis = 0)`. Standardize the array corresponding to n using the rule $x \mapsto \sqrt{n}(x - 1)$.

(b) For each n, estimate the probability that the corresponding standardized sample means from (a) are greater than 1.5. Also, find $P(Z > 1.5)$, where $Z \sim \mathcal{N}(0, 1^2)$.

(c) Display histograms of your results in (a). Create a 2×2 grid of subplots using `plt.subplots`. Let each subplot correspond to one of your standardized sample means from (a). Use 50 bins for each histogram and set the density parameter to `True`. On each subplot also overlay the graph of ϕ the probability density function of $\mathcal{N}(0, 1^2)$. Add a title and legend to each subplot.

(d) Display your results in (a) using cumulative distribution functions. Create a new 2×2 grid of subplots. Calculate the empirical cdf for each array of standardized sample means and graph it. Then overlay the graph of Φ the cumulative distribution function of $\mathcal{N}(0, 1^2)$. Add a title and legend to each subplot.