# Capstone Python Exam: Risk Modelling and Prediction

In this comprehensive assessment, you will be demonstrating your proficiency in critical financial computations using Python. This exam mirrors a real-world professional environment, integrating essential facets of teamwork, programming, version control, analysis, and documentation - **vital skills in today's fintech landscape**.

This examination can be completed individually or in teams of two. The procedure outlined below delineates your tasks:

1. **Team Formation**: Form a team of two members if you prefer. Teamwork is crucial in real-world projects, and this will give you an opportunity to enhance your collaboration skills.
2. **Git Repository**: Initiate a Git repository for your project. Here, you will commit both your code and analysis. Remember to share your Git repository URL with me. It's essential to make regular commits to document the changes and track your project's progress effectively.
3. **Coding**: Ensure that your Python code is well-structured and well-documented. It must include the requested functions according to the exercise.
4. **Analysis in PDF**: After executing your code, conduct a detailed analysis of the results. The analysis should be compiled in a professionally formatted PDF document. The document should clearly articulate your methodology, assumptions, results, and any conclusions or insights derived from the findings.
5. **Autograding**: Generate a unique file named **submission.py** that includes the primary functions to be assessed. We will employ an autograding system to evaluate these functions. Make sure to implement your functions accurately based on the specifications provided in the exercise.

This exercise aims to simulate a professional tech environment, focusing not just on coding, but also on collaboration, version control, detailed analysis, and professional presentation of results. Best of luck!

## Exercise 1: Historical Value at Risk (VaR) - Effect of Diversification

* Select two financial assets and download their historical price data.
* Calculate the daily returns for each asset.
* Construct two-asset portfolios with varying weights to examine the concept of diversification. For each weight combination, calculate the historical VaR of the portfolios.
* Compare these VaR values and write a brief explanation on the impact of diversification.
* Autograde Function: **calculate\_historical\_var(df\_portfolio\_returns, alpha)**:
  + Parameters:
    - df\_portfolio\_returns (pd.DataFrame): a DataFrame with one column containing portfolio returns.
    - alpha (float): confidence level for the VaR measure.
  + Return: VaR value (float) at the specified confidence level.
  + Example usage

df\_returns = pd.DataFrame({'returns': np.arange(-0.05, 0.06, 0.01)})

calculate\_historical\_var(df\_returns, 0.95)

This should yield -0.045.

## Exercise 2: Simulated VaR for a Two-Asset Portfolio

* Using the same two assets from Exercise 1, generate potential future price paths for these assets using a simulation based on their historical returns and volatilities.
* Assume a fixed weight for each asset in the portfolio. The weight should be inversely proportional to the asset's volatility.
* Calculate the VaR of the portfolio under different correlation assumptions between the two assets.
* Write a brief comment on how changing the correlation between the two assets affects the portfolio's VaR.
* Autograde Function: **simulated\_returns(expected\_return, volatility, correlation, numOfSim)**:
  + Parameters:
    - expected\_return (float): The expected return of the asset.
    - volatility (float): The volatility of the asset.
    - correlation (float): The correlation between the returns.
    - numOfSim (int): The number of simulations to run.
  + Return: An array of simulated returns (np.array).

## Exercise 3: EWMA Estimation of Risk (Variance)

* Choose an Exchange-Traded Fund (ETF) and download its historical price data.
* Calculate and plot the variance of ETF returns using an Exponentially Weighted Moving Average (EWMA) with decay factors of 0.94 and 0.97 over a window of 100 days.
* Reflect on your findings and how they relate to the concepts covered in the lesson on 5/16.
* Autograde Function: **calculate\_ewma\_variance(df\_etf\_returns, decay\_factor, window)**:
  + Parameters:
    - df\_etf\_returns (pd.DataFrame): A DataFrame with one column containing ETF returns.
    - decay\_factor (float): The decay factor for the EWMA (either 0.94 or 0.97).
    - window (int): The number of days in the window for the EWMA calculation (100 days in this case).
  + Returns: A DataFrame with the calculated EWMA variance values.

## Exercise 4: Machine Learning for Risk (Variance) Prediction

* Use the same asset as in Exercise 3
* Compute the daily returns and the squared returns for this asset.
* Employ a linear regression model with cross-validation to predict future variance based on lagged squared returns. For lags do not use more than 20.
* Evaluate your model's performance using the Mean Squared Error (MSE) as the error metric.
* Write a brief commentary on your model's effectiveness.