Project Week05, by Nurken Abeuov (NETID: na233)

Answer to Problem 2:

1. Calculating Value at Risk (VaR) and Expected Shortfall (ES):

Initial parameters:

- Data in *problem1.csv*, loaded as *returns*
- λ=0.97 for the Exponentially Weighted Moving Average (EWMA)
- Confidence level = 0.99

a) **Normal Distribution with Exponentially Weighted Variance (EWMA)**
   - **Exponentially Weighted Variance**:
     - The weights are calculated using λ=0.97, so that recent observations have higher weight.
     - *mean_adjusted_data = np.sqrt(weights) * (data - np.dot(weights, data))* adjusts data for variance.
     - *ew_var = np.dot(mean_adjusted_data, mean_adjusted_data)* gives the variance, and *ew_volatility = np.sqrt(ew_var)* gives the volatility.
   - **VaR Calculation**:
     - Using a Z-score for a 99% confidence level from the normal distribution, *z_score = norm.ppf(1 - confidence_level).*
     - **VaR**: is calculated as the product of the Z-score and the volatility *VaR = z_score * ew_volatility*
     - **ES Calculation**: is calculated using the formula for a normal distribution: $ES = (pdf(z_{score}) / (1-\alpha)) \times ew\_volatility$

b) **MLE Fitted T-Distribution**
   - **Fitting the T-Distribution Parameters**:
     - A T-distribution is fitted to the data using MLE with *scipy.optimize.minimize* to find optimal values for μ, σ, and ν.
   - **VaR Calculation**:
     - **VaR**: is calculated using the quantile function for the T-distribution with a 99% confidence level: $VaR = \mu + \sigma \times t.ppf\,(1-\alpha, \nu)$
   - **ES Calculation**:
     - **ES**: under the T-distribution uses formula: $ES = \mu + \sigma \times (pdf\,(t_{quantile}) / (1-\alpha)) \times (\nu + t_{quantile}^{2}) / (\nu - 1)$

c) **Historical Simulation**
   - **VaR Calculation**:
     - This method sorted historical returns and picked the 1st percentile (for 99% confidence): *VaR = sorted_data[var_index]*
   - **ES Calculation**:
     - It is the mean of all returns below this VaR threshold: *ES = np.mean(sorted_data[:var_index])*

Output of VaR & ES Across Distributions:

#	Probabilistic Distributions	VaR	ES
1	Normal Distr. with EWMA	0.1277	0.1463
2	MLE Fitted T-Distr.	0.1265	0.1671
3	Hist. Simulation	0.1459	0.1838

2. **Comparison of VaR and ES Across Distributions:**

a) **Normal Distribution with EWMA**:
   - Normality implies a symmetric distribution of returns, so both VaR and ES are driven by standard deviations away from the mean.
   - VaR is smaller (0.1277), and ES is also relatively lower (0.1463) due to the symmetry and moderate tail risk under normal distribution assumptions.

b) **MLE Fitted T-Distribution**:
   - The T-distribution assumes heavier tails, resulting in a slightly higher ES than the normal distribution, highlighting potential extreme losses.
   - Higher ES (0.1671) than EWMA (0.1463), indicating greater sensitivity to extreme losses, while VaR (0.1265) is similar due to moderate tail weight.

c) **Historical Simulation**:
   - VaR reflects actual historical returns and often captures extreme losses better than parametric methods. The largest ES across methods, as it is based directly on extreme observed losses, making it sensitive to past extreme events.
   - This method shows the highest VaR (0.1459) and ES (0.1838), as it fully accounts for historical tail behavior without distributional assumptions.

Conclusion:

- **Normal EWMA** provides more stable, moderate-risk estimate assuming normally distributed returns.
- **MLE T-Distribution** adjusts for tail risk with higher ES sensitivity to extremes.
- **Historical Simulation** best captures historical losses, providing the highest ES by directly measuring past tail behavior.

Answer to Problem 3:

1. Importing Required Libraries and Setting Up Paths

- The code imports various functions from the my risk management library (***risk_management_lib***) and loads the portfolio and price data for further analysis.

2. Calculating Arithmetic Returns

- Using *return_calculate* function, the code calculates **arithmetic returns** for each stock using daily prices from DailyPrices.csv. This step produces a DataFrame *(returns)* where each column represents a stock, and each row represents the daily return for that stock.

3. Portfolio Separation

- The portfolio data (*portfolio.csv*) is split into three subsets *(portfolio_a, portfolio_b, portfolio_c)* based on the columns in *Portfolio*, corresponding to the three individual portfolios (A, B, and C).

4. Calculating VaR and ES for Portfolios A and B with Generalized T-distribution

For **portfolios A and B**, the code fits a generalized T-distribution to each stock's returns in the respective portfolios. Here's the step-by-step process:

1. **Loop through each stock in the portfolio**:
   - For each stock in Portfolio A, *fit_generalized_t* is called on the stock's returns to fit a generalized T-distribution model. This function returns *mu* (mean), *sigma* (scale), and *nu* (degrees of freedom), which parameterize the fitted T-distribution.
2. **Calculate VaR and ES using T-distribution**:
   - *var_t* and *es_t* functions are used to compute the Value at Risk (VaR) and Expected Shortfall (ES) based on the fitted parameters for each stock, at a 99% confidence level. These values are stored in *VaRs_A* and *ESs_A* for Portfolio A and *VaRs_B* and *ESs_B* for Portfolio B.
3. **Aggregate VaR and ES for Portfolios A and B**:
   - The total VaR and ES for Portfolios A and B are calculated by summing up the individual *VaRs_A* and *ESs_A* for each stock in Portfolio A and *VaRs_B* and *ESs_B* for each stock in Portfolio B.

5. Calculating VaR and ES for Portfolio C with Normal Distribution

For **Portfolio C**, the code assumes a **Normal distribution** for the stock returns:

1. **Loop through each stock in Portfolio C**:
   - For each stock, the code calculates the mean (*mu*) and standard deviation (*sigma*) of the returns.
2. **Calculate VaR and ES using Normal distribution**:
   - *var_normal* and *es_normal* functions are used to compute the VaR and ES at the 99% confidence level for each stock using the Normal distribution parameters.
3. **Aggregate VaR and ES for Portfolio C**:
   - The VaR and ES for Portfolio C are calculated by summing the individual VaR and ES values across all stocks in the portfolio.

6. Combining Portfolios A, B, and C with a Copula-Based Approach

To obtain a **combined VaR and ES** for all portfolios:

1. **Calculate Portfolio Returns for A, B, and C**:
   - The function *calculate_portfolio_returns* computes weighted returns for each portfolio based on each stock's *asset value (price * holdings*).
2. **Construct Correlation Matrix**:
   - The aggregated returns for each portfolio are stored in a DataFrame (*portfolio_agg*), from which a correlation matrix *(correlation_matrix)* is computed to understand the relationships between portfolios A, B, and C.
3. **Copula-Based VaR and ES Calculation**:
   - Using the function *copula_based_var_es*, a copula model is fitted to the correlation matrix and used to simulate the joint distribution of returns. This provides an estimate for combined VaR and ES at the 99% confidence level.

7. Comparison with EWMA-Based VaR from Week 4

The code calculates **EWMA-based VaR and ES** for each portfolio to compare with the previous approach. This step utilizes an exponentially weighted moving average (EWMA) model to calculate volatility.

1. **Calculate Portfolio Returns**:

- *calculate_portfolio_ewma_var* uses *calculate_portfolio_returns* to get portfolio returns for each of A, B, and C.

2. **EWMA-Based VaR and ES Calculation**:
   - *var_normal* and es_normal are applied to the EWMA-calculated volatility to compute VaR and ES.
3. **Sum Total EWMA VaR and ES**:
   - Finally, the total EWMA VaR and ES values are obtained by summing the individual values for portfolios A, B, and C.

8. Final Output:

EWMA (Exponentially Weighted Moving Average) Method

#	Portfolio	VaR	ES
1	Portfolio A	0.0220	0.0252
2	Portfolio B	0.0181	0.0207
3	Portfolio C	0.0182	0.0209
4	Total Portfolio (EWMA)	0.0583	0.0668

Combined Approach (Copula and Distributional Assumptions)

#	Portfolio	VaR	ES
1	Portfolio A	1.5824	2.2568
2	Portfolio B	1.1850	1.7189
3	Portfolio C	1.2334	1.4130
4	Total Combined (Copula-Based)	11.2044	14.6615

Conclusion:

The combined approach reveals higher potential losses than the EWMA method, indicating that incorporating fat-tail distributions and copula-based dependencies provides a more robust assessment of extreme risks. The EWMA method remains useful for portfolios with normal-like returns but is less effective in capturing extreme events.