**Case:**

This assignment is designed to introduce you to a simplified method for portfolio allocation using mean variance analysis.

* Many hedge funds use these factors in developing their key portfolios marketed to investors.
* From a consulting/marketing perspective, asset managers who have a clear view on the different asset classes will not only be able to design better (i.e., more profitable) portfolios, but also sell ETF type of products which for example may hedge inflation risks or declining growth risks.
* This approach is therefore at the of practical asset management with implications for product design and business development.

**Directions:**

The “IBOV” sheet is populated with daily prices for the Brazilian Ibovespa Index, while the “Data” sheet is populated with nominal return data. You will be asked to create a portfolio allocation table. Read all of the directions and questions before starting, as what you will be required to do may shape the best way to organize your responses.

In financial markets, it is well known that real and nominal mean variance analysis often produce similar allocations within the time period we are analyzing. Since we arguably don’t have a good proxy for the real risk-free, and mean variance analysis can be sensitive to the risk-free rate we will conduct all analysis for this assignment in **nominal terms**. Keep in mind that this approximates the real mean variance analysis, particularly built for this exercise.

1. Follow the instructions below to solve the case. Do not forget to submit a document with your code and the respective outputs!
   1. Construct a series of daily returns (linear or exponential) for each asset. The series of returns for each asset should be in a single dataframe and the answer must be an SQL query (or “a” SQL query if you call it sequel) but it may be coded in any language that can process it. Utilizing more complex ways of querying is preferred.
   2. Calculate the mean daily nominal return, standard deviation of daily returns, and monthly Sharpe Ratio (consider a month as 30 days) for each asset using only the data on that sheet. You may assume that the mean return on the TBill is a good proxy for the nominal risk-free rate.
   3. Calculate the variance-covariance matrix (VCV) for the four risky assets’ nominal returns (equity, bonds, gold, and commodities) using only the data on that sheet like the example below.

Text, letter

Description automatically generated

* 1. Use elements you have already calculated to calculate the portfolio weights, expected return, expected standard deviation of returns, and expected Sharpe Ratio of the maximal Sharpe Ratio portfolio, using both the formulas provided below and Excel’s solver. **Assume that the weights must be positive, so no short selling.**
  2. Repeat step c for the optimal portfolio allocations for the Global Minimum Variance portfolio, using both the formulas provided below and Excel’s solver. **Assume that the weights must be positive, so no short selling.**

1. Create a summary table which contains: the mean, standard deviation, and Sharpe Ratio of each asset return and the weights of the portfolio optimal for a mean-variance investor.
2. Solve both exercises above on a programming language of your choice (Preferably in Python, formulas and optimization, but you can use any language you desire as long as you have a deliverable that contains your code and the expected output. A PDF document should suffice.).

**Additional resources:**

**Portfolio Math with Many Assets:**

Notation: **w** a vector of portfolio weights, **E(R)** a vector of their expected returns, **Ω** their variance-covariance matrix (VCV), and rf the risk-free rate.

Expected Return E(rp)= **E(R)**

Expected Variance σ2p= **w’ Ω w**

Sharpe Ratio = (E(rp)-rf)/σp

Remember that for the purposes of this assignment you will use historical means or sample standard deviations of real returns to estimate each of the inputs into these equations. These can be implemented in Excel as follows:

Expected Return: =MMULT([returns vector], [weights vector])

Expected Standard Deviation:

=(MMULT([weights vector],MMULT([VCV matrix],TRANSPOSE([weights vector]))))^0.5

Both of these formulas must be entered as single cell array formulas to work properly. A single cell array formula is entered by editing the formula by clicking into the cell or with F2 then exiting the cell by clicking Ctrl + Shift + Enter (Windows default). This will add curly brackets to the outside of the formula until you edit the formula again.

**Unconstrained Allocation Problem Solutions with Many Assets:**

Notation: **w** a vector of portfolio weights exclusive of the risk-free asset, **1** a vector of ones, **E(R)** a vector of their expected returns, **Ω** their variance-covariance matrix (VCV) of risky asset returns and rf the risk-free rate (a scalar number).

We consider here the Maximal Sharpe Ratio (MSR) (Mean Variance Efficient (MVE), Tangency) Portfolio, the Global Minimum Variance (GMV) Portfolio, and portfolios for mean-variance investors with a given risk aversion, A.

The MSR Allocation: **wMSR = Ω-1(E(R)-**rf**1)/(1’Ω-1(E(R)-**rf**1))**

The GMV Allocation: **wGMV = Ω-11/(1’Ω-11)**

All of these optimizations are subject to the unit cost portfolio constraint that **w’1+**wrf = 1.

**Excel Implementation with Array Formulas:**

All of these formulas are straightforward to implement in Excel either using matrix formulas and multi-cell arrays or the Solver. These formulas return column vectors of the weights on the risky assets. The results can be transposed by surrounding the equation with TRANSPOSE() to return row vectors and the sum of the weights should amount to 1.

These formulas must be entered as multi-cell (range) array formulas. To enter a multi-cell array formula highlight all cells that will be part of the output (in this case a column vector of 4 cells) and press F2 (Windows default) to enter the formula editor and enter the formula. Press Ctrl + Shift + Enter (Windows default) to exit the cell and enter the formula as an array formula. The array formula will now be entered in all cells that were originally highlighted. Note that once this is done you can no longer edit part of the array without editing all of it. This includes cell operations like insertion or deletion that may move cells to break up the array. Attempts to break up or alter the array will throw errors.

The MSR Allocation:

=TRANSPOSE(MMULT(MINVERSE([VCV matrix]);TRANSPOSE([returns row vector])-[vector of 1]\*rf)/MMULT(TRANSPOSE([returns row vector]);MMULT(MINVERSE([VCV matrix]);TRANSPOSE([returns row vector])-[vector of 1]\*rf)))

The GMV Allocation:

=TRANSPOSE(MMULT(MINVERSE([VCV matrix]);[vector of 1])/MMULT(TRANSPOSE([vector of 1]); MMULT(MINVERSE([VCV matrix]);[vector of 1])))