Lab 7: Portfolio Optimization

Learning Objectives:

- Transform non-linear constraints into linear constraints using auxiliary variables. (Analyze)
- Write code on paper to implement an optimization formulation. (Code)

Problem Description

Shanice is an analyst at Trojan Capital Management, which has recently been exploring optimization-based techniques for portfolio management. Since Shanice received training in optimization as part of her Master's degree, she has been given the task of implementing a prototype of the following optimization in Python.

Input Data:

- S: the set of stocks.
- w_i: the old weight of stock $i \in S$ before optimization. (The "weight" of a stock is % of total funds invested in the stock; weights of all stocks should add to one.)
- R_i: the expected annual return of stock $i \in S$.
- C_{ij}: the estimated covariance between stocks $i, j \in S$.
- σ_{target}: the maximum volatility of the final portfolio.
- δ: the maximum total change allowed between the old weights and the new weights.
- k: the maximum # of stocks allowed in the portfolio.
- ϵ: the minimum non-zero weight allowed.

Decision variables:

- x_i: the new weight of stock i. (Continuous)

Formulation: All summations are over the set S of stocks.

$$
\begin{array}{rcl}
\text{Maximize:} & \displaystyle\sum_i R_i x_i & \text{(Average Return)} \\
\text{subject to:} & & \\
\text{(Valid weights)} & \displaystyle\sum_i x_i = 1 & \\
\text{(Risk tolerance)} & \displaystyle\sum_i \sum_j C_{ij} x_i x_j \leq \sigma_{target}^2 & \\
\text{(Change in weights)} & \displaystyle\frac{1}{2}\sum_i |x_i - w_i| \leq \delta & \\
\text{(Simplicity)} & (\#\text{ of stock } i \text{ with } x_i > 0) \leq k & \\
\text{(Non-negligible weights)} & \text{If } x_i > 0 \text{ then } x_i \geq \epsilon & \text{for each stock } i \in S. \\
 & x_i \geq 0 &
\end{array}
$$

However, the last three constraints are not allowed in Gurobi, as they are not linear. (The risk tolerance constraint, on the other hand, is allowed in Gurobi because the LHS can be expressed as a sum of squares.)

Exercise A. (15 minutes) Use auxiliary decision variables to rewrite the last three constraints in a linear way. Then rewrite the entire formulation on another sheet of paper.

Lab Deliverable: Code on Paper

Exercise B. (25 minutes) Write your name on a blank piece of paper and write code on this piece of paper to implement the formulation from part a). (To be handed in at the end of the class.) The code should output the final portfolio in an Excel sheet named `portfolio.xlsx` where the first column is the name of the stock and the second column is the corresponding weight. The column headers are `Stock` and `Weight`. The output file should only contain stocks with positive weights.

You may assume that the inputs are supplied in the following formats:

- w_i: A DataFrame named `oldPortfolio` with row index being the stock i and a column named `Weight` containing the w_i. (see below)
- R_i: A Series named `ret` with mapping each stock i to its expected annual return R_i. (see below)
- C_{ij}: A DataFrame named `cov` with row index being the i's and column index being the j's. (see below)
- σ_{target}: a float variable named `stdMax`.
- δ: a float variable named `maxChange`.
- k: an integer variable named `k`.
- ϵ: a float variable named `eps`.

```
[1]: oldPortfolio.head()
```

```
        Weight
Stock
AMGN  0.306342
CNC   0.231379
FFIV  0.290586
FL    0.019480
LEG   0.152214
```

```
[2]: ret.head()
```

```
MMM     0.146843
AOS     0.365267
ABT     0.122198
ACN     0.204758
ATVI    0.229294
dtype: float64
```

```
[3]: cov.head()
```

```
           MMM       AOS       ABT       ACN      ATVI       AYI      ADBE
MMM   0.071051  0.061622  0.030693  0.045073  0.045842  0.063481  0.058512
AOS   0.061622  0.143255  0.037418  0.057101  0.068267  0.093367  0.080927
ABT   0.030693  0.037418  0.061038  0.033389  0.033699  0.038779  0.039315
ACN   0.045073  0.057101  0.033389  0.092509  0.051237  0.063701  0.062303
ATVI  0.045842  0.068267  0.033699  0.051237  0.175176  0.068245  0.079826
[5 rows x 450 columns]
```