Certificate in Quantitative Finance

Final Project Brief

January 2023 Cohort

This document outlines topics available for this cohort. No other topics can be submitted. Each topic has by-step instructions to give you a structure (not limit) as to what and how to implement.

Marks earned will strongly depend on your coding of numerical techniques and presentation of how you explored and tested a quantitative model (report in PDF or HTML). Certain numerical methods are too involved or auxiliary to the model, for example, do not recode optimisation or RNs generation. Code adoption allowed if the code fully modified by yourself.

A capstone project requires own study and ability to work with documentation on packages that implement numerical methods in your coding environment e.g., Python, R, Matlab, C#, C++, Java. You do not need to pre-approve the coding language and use of libraries, including very specialised tools such as Scala, kdb+ and q. However, software like EViews is not coding.

Exclusively for current CQF delegates. No distribution.

To complete the project, you must code the model(s) and its numerical techniques form one topic from the below options and write an analytical report. If you continue from a previous cohort, please review topic description because tasks are regularly reviewed. It is not possible to submit past topics.

1. Credit Spread for a Basket Product (CR)
2. Deep Learning for Financial Time Series (DL)
3. Pairs Trading Strategy Design & Back test (TS)
4. Portfolio Construction using Black-Litterman Model and Factors (PC)
5. Optimal Hedging with Advanced Greeks (DH)

Topics List for the current cohort will be available on the relevant page of Canvass Portal.

Project Report and Submission

- First recommendation: do not submit Python Notebook ' as is' – there is work to be done to transform it into an analytical report. Remove printouts of large tables/output. Write up mathematical sections (with LaTeX markup). Write up analysis and comparison for results and stress-testing (or alike). Explain your plots. Think like a quant about the computational and statistical properties: convergence/accuracy/variance and bias. Make a table of the numerical techniques you coded/utilised.
- Project Report must contain sufficient mathematical model(s), numerical methods and an adequate conclusion discussing pros and cons, further development.
- There is no set number of pages. Some delegates prefer to present multiple plots on one page for comparability, others choose more narrative style.
- It is optimal to save Python Notebook reports as HTML but do include a PDF with page numbers – for markers to refer to.
- Code must be submitted and working.

FILE 1. For our download and processing scripts to work, it is necessary to name and upload the project report as ONE file (pdf or html) with the two-letter project code, followed by your name as registered on CQF Portal.
Examples: TS John Smith REPORT.pdf or PC Xiao Wang REPORT.pdf

FILE 2. All other files, code and a pdf declaration (if not the front page) must be uploaded as additional ONE zip file, for example TS John Smith CODE.zip. In that zip include converted PDF, Python, and other code files. Do not submit unzipped .py, .cpp files as cloud anti-virus likely to flash red on our side.
Do not submit files with generic names, such as CODE.zip, FinalProject.zip, Final Project Declaration.pdf, etc. Such files will be disregarded.

Submission date for the project is Monday 21st August 2023, 23.59 BST

There is no extension time to Final Project.

Projects without a hand-signed declaration or working code are incomplete.

Failure to submit ONE report file and ONE zip file according to the naming instructions means such a project will miss an allocation for grading.

All projects are checked for originality. We reserve an option of a viva voce before the qualification to be awarded.

Project Support

<u>Advanced Electives</u>

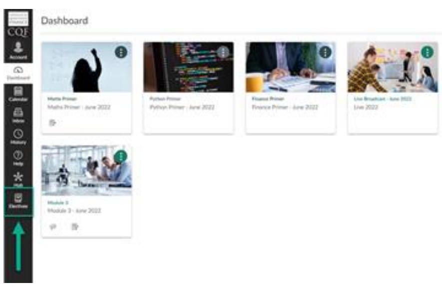
To gain background knowledge in a focused way, we ask you to review two Advanced Electives. Electives canvass knowledge areas and can be reviewed before/at the same time/closer to writing up Analysis and Discussion (explanation of your results).

- There is no immediate match between Project Topics and Electives
- Several workable combinations for each Project Topic are possible
- One elective learning strategy is to select one `topical elective' and one `coding elective'

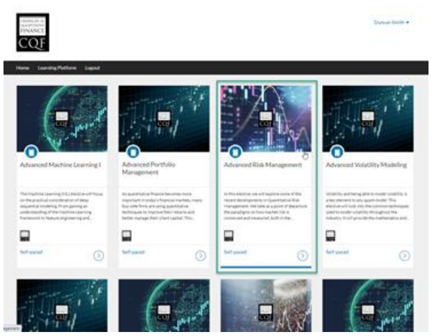
To access the electives:

Login to the CQF Learning Hub

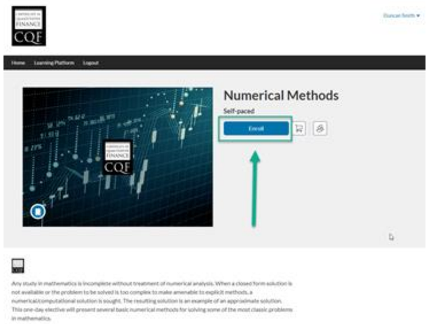
Click the *Learning Platform* button to sign into Canvas



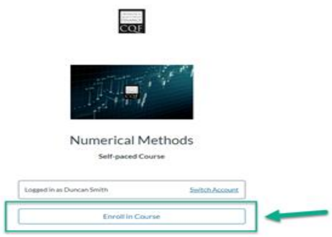
Click on *Electives* button on global navigation menu



You will be redirected to the electives Catalogue, where you can view and review all electives available to you. Full descriptions for each elective can be found here.

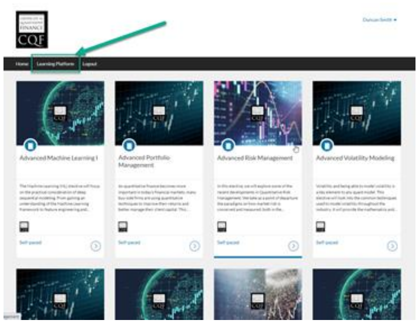


When on an elective click the *enrol* button



You will see the confirmation page, click the *enrol in Course* button to confirm your selection

You will land on the successful enrolment page, where you can click to start the elective or return to the catalogue page



When on the catalogue page you can click the *Learning Platform* link to return to Canvas. Your electives selected will appear on your learning dashboard

Workshop & Tutorials

Each project title is supported by a faculty member alongside a set of project workshops and tutorials.

DATE	TITLE	TIME
01/07/2023	Final Project Workshop I	13:00 – 15:30 BST
08/07/2023	Final Project Workshop II	13:00 – 15:30 BST
11/07/2023	Final Project Tutorial I	18:00 – 19:00 BST
12/07/2023	Final Project Tutorial II	18:00 – 19:00 BST
13/07/2023	Final Project Tutorial III	18:00 – 19:00 BST
14/07/2023	Final Project Tutorial IV	18:00 – 19:00 BST

Faculty Support

Title: Credit Spread for a Basket Product
Project Code: CR
Lead: Riaz Ahmad

Title: Deep Learning for Financial Time Series
Project Code: DL
Lead: Kannan Singaravelu

Title: Pairs Trading Strategy Design & Backtest
Project Code: TS
Faculty Lead: Richard Diamond

Title: Portfolio Construction using Black-Litterman Model and Factors
Project Code: PC
Faculty Lead: Panos Paras

Title: Optimal Hedging with Advanced Greeks
Project Code: DH
Faculty Lead: Richard Diamond

To ask faculty a question on your chosen topic, please submit a support ticket by clicking on the Support button which can be found in the bottom hand right corner on your portal.

Coding for Quant Finance

- Choose programming environment that has appropriate strengths and facilities to implement the topic (pricing model). Common choice is Python, Java, C++, R, Matlab. Exercise judgement as a quant: which language has libraries to allow you to code faster, validate easier.

- Use of R/Matlab/Mathematica is encouraged. Often there a specific library in Matlab/R gives fast solution for specific models in robust covariance matrix/cointegration analysis tasks.

- Project Brief give links to nice demonstrations in Matlab, and Webex sessions demonstrate Python notebooks {does not mean your project to be based on that ready code

- Python with pandas, matplotlib, sklearn, and tensorow forms a considerable challenge to Matlab, even for visualization. Matlab plots editor is clunky, and it is not that difficult to learn various plots in Python.

- 'Scripted solution' means the ready functionality from toolboxes and libraries is called, but the amount of own coding of numerical methods is minimal or non-existent. This particularly applies to Matlab/R.

- Projects done using Excel spreadsheet functions only are not robust, notoriously slow and do not give understanding of the underlying numerical methods. CQF-supplied Excel spreadsheets are a starting point and help to validate results but coding of numerical techniques/use of industry code libraries is expected.

- The aim of the project is to enable you to code numerical methods and develop model prototypes in a production environment. <u>Spreadsheets-only or scripted solutions are below the expected standard for completion of the project.</u>

- What should I code? Delegates are expected to re-code numerical methods that are central to the model and exercise judgement in identifying them. Balanced use of libraries is at <u>own discretion</u> as a quant.

- Produce a small table in report that lists methods you implemented/adjusted. If using ready functions/borrowed code for a technique, indicate this and describe the limitations of numerical method implemented in that code/standard library.

- It is up to delegates to develop their own test cases, sensibility checks and validation. It is normal to observe irregularities when the model is implemented on real life data. If in doubt, reflect on the issue in the project report.

- The code must be thoroughly tested and well-documented: each function must be described, and comments must be used. Provide instructions on how to run the code.

Credit Spread for a Basket Product

Price a fair spread for a portfolio of CDS for 5 reference names (Basket CDS), as an expectation over the joint distribution of default times. The distribution is unknown analytically and so, co-dependent uniform variables are sampled from a copula and then converted to default times using a marginal term structure of hazard rates (separately for each name). Copula is calibrated by estimating the appropriate default correlation (historical data of CDS differences is natural candidate but poses market noise issue). Initial results are histograms (uniformity checks) and scatter plots (co-dependence checks). Substantial result is sensitivity analysis by repricing.

A successful project will implement sampling from both, Gaussian and t copulae, and price all k-th to default instruments (1st to 5th). Spread convergence can require the low discrepancy sequences (e.g., Halton, Sobol) when sampling. Sensitivity analysis *wrt* inputs is required.

Data Requirements

Two **separate** datasets required, together with matching discounting curve data for each.

1. **A snapshot of credit curves** on a particular day. A debt issuer likely to have a USD/EUR CDS curve – from which a term structure of hazard rates is bootstrapped and utilised to obtain exact default times, $u_i \to \tau_i$. In absence of data, spread values for each tenor can be assumed or stripped visually from the plots in financial media. The typical credit curve is concave (positive slope), monotonically increasing for $1Y, 2Y, \ldots, 5Y$ tenors.

2. **Historical credit spreads time series** taken at the most liquid tenor $5Y$ for each reference name. Therefore, for five names, one computes 5×5 default correlation matrix. Choosing corporate names, it is much easier to compute correlation matrix from equity returns.

   Corporate credit spreads are unlikely to be in open access; they can be obtained from Bloomberg or Reuters terminals (via your firm or a colleague). For sovereign credit spreads, time series of ready bootstrapped PD_{5Y} were available from DB Research, however, the open access varies. Explore data sources such as `www.datagrapple.com` and `www.quandl.com`.

   Even if CDS_{5Y} and PD_{5Y} series are available with daily frequency, the co-movement of daily changes is market noise *more* than correlation of default events, which are rare to observe. Weekly/monthly changes give more appropriate input for default correlation, however that entails using 2-3 years of historical data given that we need at least 100 data points to estimate correlation with the degree of significance.

If access to historical credit spreads poses a problem remember, default correlation matrix can be estimated from historic equity returns or debt yields.

Step-by-Step Instructions

1. For each reference name, bootstrap implied default probabilities from quoted CDS and convert them to a term structure of hazard rates, $\tau \sim Exp(\hat{\lambda}_{1Y}, \ldots, \hat{\lambda}_{5Y})$.

2. Estimate default correlation matrices (near and rank) and d.f. parameter (ie, calibrate copulæ). You will need to implement pricing by Gaussian and t copulæseparately.

3. Using sampling form copula algorithm, repeat the following routine (simulation):

   (a) Generate a vector of correlated uniform random variable.

   (b) For each reference name, use its term structure of hazard rates to calculate exact time of default (or use semi-annual accrual).

   (c) Calculate the discounted values of premium and default legs for every instrument from *1st* to *5th*-to-default. Conduct MC separately or use one big simulated dataset.

4. Average premium and default legs across simulations separately. Calculate the fair spread.

Model Validation

- The fair spread for k*th*-to-default Basket CDS should be less than k-1 to default. Why?

- Project Report on this topic should have a section on **Risk and Sensitivity Analysis** of the fair spread *w.r.t.*

  1. default correlation among reference names: either stress-test by constant high/low correlation or ± percentage change in correlation from the actual estimated levels.
  2. credit quality of each individual name (change in credit spread, credit delta) as well as recovery rate.

  Make sure you discuss and compare sensitivities for all five instruments.

- Ensure that you explain historical sampling of default correlation matrix and copula fit (uniformity of pseudo-samples) – that is, Correlations Experiment and Distribution Fitting Experiment as will be described at the Project Workshop. Use histograms.

Copula, CDF and Tails for Market Risk

The recent practical tutorial on using copula to generate correlated samples is available at:
`https://www.mathworks.com/help/stats/copulas-generate-correlated-samples.html`
Semi-parametric CDF fitting gives us percentile values with fitting the middle and tails. Generalised Pareto Distribution applied to model the tails, while the CDF interior is Gaussian kernel-smoothed. The approach comes from Extreme Value Theory that suggests correction for an Empirical CDF (kernel fitted) because of the tail exceedances.

`http://uk.mathworks.com/help/econ/examples/using-extreme-value-theory-and-copulas-to-evaluate-market-risk.html`

`http://uk.mathworks.com/help/stats/examples/nonparametric-estimates-of-cumulative-distribution-functions-and-their-inverses.html`

Reading List:

- Very likely you will revisit *CDO & Copula Lecture* material, particularly slides 48-52 that illustrate Elliptical copula densities and discuss Cholesky factorisation.
- *Sampling from copula* algorithm is in *relevant Workshop* and *Monte Carlo Methods in Finance* textbook by Peter Jaekel (2002) – see Chapter 5.
- Rank correlation coefficients are introduced *Correlation Sensitivity Lecture* and P. Jaekel (2002) as well. CR Topic Q&A document gives the clarified formulae and explanations.

Deep Learning for Financial Time Series

Summary

Trend prediction has drawn a lot of research for many decades using both statistical and computing approaches including machine learning techniques. Trend prediction is valuable for investment management as accurate prediction could ensure asset managers outperform the market. Trend prediction remains a challenging task due to the semi-strong form of market efficiency, high noise-to-signal ratio, and the multitude of factors that affect asset prices including, but not limited to the stochastic nature of underlying instruments. However, sequential financial time series can be modeled effectively using sequence modeling approaches like a recurrent neural network.

Objective

Your objective is to produce a model to predict positive moves (up trend) using the Long Short-Term Memory Networks. Your proposed solution should be comprehensive with the detailed model architecture, evaluated with a backtest applied to a trading strategy.

- Choose one ticker of your interest from the index, equity, ETF, crypto token, or commodity.
- Predict trend only, for a short-term return (example: daily, 6 hours). Limit prediction to binomial classification: the dependent variable is best labelled [0, 1]. Avoid using [-1, 1] as class labels.
- Analysis should be comprehensive with detailed feature engineering, data pre-processing, model building, and evaluation.

Note: You are free to make study design choices to make the task achievable. You may redefine the task and predict the momentum sign (vs return sign) or direction of volatility. Limit your exploration to ONLY one asset. At each step, the process followed should be expanded and explained in detail. Merely presenting python codes without a proper explanation shall not be accepted. The report should present the study in a detailed manner with a proper conclusion. Code reproducibility is a must and the use of modular programming approaches is recommended. Under this topic, you do not recode existing indicators, libraries, or optimization to compute neural network weights and biases.

Step-by-Step Instructions

1. The problem statement should be explicitly specified without any ambiguity including the selection of underlying assets, datasets, timeframe, and frequency of data used.

- If predicting short-term return signs (for the daily move), then training and testing over up to 5 years should be sufficient. If you attempt the prediction of 5D, 10D return for equity or 1W, 1M for the Fama French factor, you'll have to increase the data required to at least 10 years.

2. Perform exhaustive Feature Engineering (FE).

- FE should be detailed including the listing of derived features and specification of the target/label. Devise your approach on how to categorize extremely small near-zero returns (drop from the training sample or group with positive/negative returns). The threshold will strongly depend on your ticker. Example: small positive returns below 0.25% can be labelled as negative.
- Class imbalances should be addressed - either through model parameters or via label definition.
- Use of features from cointegrated pairs and across assets is permitted but should be tactical about design. There is no one recommended set of features for all assets; however, the initial feature set should be sufficiently large. Financial ratios, advanced technical indicators including volatility estimators, and volume information can be a predictor for price direction.
- OPTIONAL Use of news heatmap, credit spreads (CDS), historical data for financial ratios, history of dividends, purchases/disposals by key stakeholders (director dealings) or by large funds, or Fama-French factor data can enhance your prediction and can be sourced from your professional subscription.

3. Conduct a detailed Exploratory Data Analysis (EDA).

- EDA helps in dimensionality reduction via a better understanding of relationships between features and uncovers underlying structure, and invites detection/explanation of the outliers. The choice of feature scaling techniques should be determined by EDA.

4. Proper handling of data is a must. The use of a different set of features, lookback length, and datasets warrant cleaning and/or imputation.

5. Feature transformation should be applied based on EDA.

- Multi-collinearity analysis should be performed among predictors.
- Multi-scatter plots presenting relationships among features are always a good idea.
- Large feature sets (including repeated kinds, and different lookbacks) warrant a reduction in dimensionality in features. Self Organizing Maps (SOM), K-Means clustering, or other methods can be used for dimensionality reduction. Avoid using Principal Component Analysis (PCA) for non-linear datasets/predictors.

6. Perform extensive and exhaustive model building.

- Design the neural network architecture after extensive and exhaustive study.
- The best model should be presented only after performing the hyperparameter optimization and compared with the baseline model.
- The choice and number of hyperparameters to be optimized for the best model are design choices. Use experiment trackers like MLFlow or TensorBoard to present your study.

7. The performance of your proposed classifier should be evaluated using multiple metrics including backtesting of the predicted signal applied to a trading strategy.

- Investigate the prediction quality using AUC, confusion matrix, and classification report including balanced accuracy (if required) .
- Predicted signals should be evaluated by applying them to a trading strategy.

* * *

Pairs Trading Strategy Design & Backtest

Estimation of a cointegrated relationship between prices allows to arbitrage the mean-reverting spread, known as special 'cointegrated residual'. Put trade design and backtesting at the centre of the project, think about signal generation from OU process and $P\&L$ backtesting from the beginning. Pairs Trading was conventionally done using correlation and you can still correlate assets in search of co-moving pairs. However, using 100% -100% weights is naive and cointegration analysis is more appropriate and robust for non-stationary series, which asset prices are. Signal generation and suitability of the cointegrated residual for trading depend on its fitting to OU process and solution to its SDE, which is essentially the same as Vasicek model in rates.

The numerical techniques to implement: regression computation in matrix form, Engle-Granger procedure, and statistical tests. You are encouraged to venture into A) multivariate cointegration (VECM, Johansen procedure) and B) robustness checking of cointegration weights, ie, by adaptive estimation of your regression parameters, however the latter is not a requirement. Advantage of multivariate cointegration/Johansen is the weights of your trading strategy will be difficult to guess from the outside. That comes, however, with the loss of $P\&L$ attribution (explanation), absence of good Python libraries (to year 2023) and in comparison, Engle-Granger procedure is very explicit, error correction model.

Signal Generation and Backtesting

- Be inventive beyond equity pairs: consider commodity futures, instruments on interest rates, and aggregated indices.
- Arb is realised by using cointegrating coefficients $\boldsymbol{\beta}_{Coint}$ as allocations $\boldsymbol{w}$. That creates a long-short portfolio that generates a mean-reverting spread. All project designs should include trading signal generation (from OU process fitting) and backtesting (drowdown plots, rolling SR, rolling betas).
- Does cumulative P&L behave as expected for a cointegration arb trade? Is P&L coming from a few or many trades, what is half-life? Maximum Drawdown and behaviour of volatility/VaR?
- Introduce liquidity and algorithmic flow considerations (a model of order flow). Any rules on accumulating the position? What impact bid-ask spread and transaction costs will make?

Step-by-Step Instructions

Can utilise the ready multivariate cointegration (R package *urca*) to identify your cointegrated cases first, especially if you operate with the system such as four commodity futures (of different expiry but for the period when all traded. 2-3 pairs if analysing separate pairs by EG.

Part I: Pairs Trade Design

1. Even if you work with pairs, re-code regression estimation in matrix form – your own OLS implementation which you can re-use. Regression between stationary variables (such as DF test regression/difference equations) has OPTIONAL model specification tests for (a) identifying optimal lag p with AIC BIC tests and (b) stability check.

2. Implement Engle-Granger procedure for each your pair. For Step 1 use Augmented DF test for unit root with lag 1. For Step 2, formulate both correction equations and decide which one is more significant.

3. Decide signals: common approach is to enter on bounds $\mu_e \pm Z\sigma_{eq}$ and exit on e_t everting to about the level μ_e.

4. At first, assume $Z = 1$. Then change Z slightly upwards and downwards – compute P&L for each case of widened and tightened bounds that give you a signal. Alternatively run an optimisation that varies Z_{opt} for $\mu_e \pm Z_{opt}\sigma_{eq}$ and either maximises the cumulative P&L or another criterion.

   Caution of the trade-off: wider bounds might give you the highest P&L and lowest N_{trades} however, consider the risk of co-integration breaking apart.

5. OPTIONALLY attempt multivariate cointegration with R package *urca* – as of 2023 Python VECM models are only available in Github dev versions of *statsapi* – in order select the best candidates for pairs/basket trading.

Part II: Backtesting

It is your choice as a quant to decide which elements you need to present on the viability, robustness and 'uncorrelated returns' nature of your trading strategy.

4. Think of machine learning-inspired backtesting, such as splitting data into train/test subsets, preprocessing, and crossvalidaiton as appropriate and feasible (beware of crossvalidation issues with time series analysis).

5. Perform systematic backtesting of your trading strategy (returns from a pairs trade): produce drawdown plots, rolling Sharpe Ratio, at least one rolling beta *wrt* to the S&P500 excess returns. However discuss why rolling beta(s) might not be as relevant to stat arb and market-making.

6. OPTIONALLY Academic research will test for breakouts in cointegrated relationship with LR test. Cointegrated relationship supposed to persist and β'_{Coint} should stay the same: continue delivering the stationary spread over 3-6 months without the need to be updated. Is this realistic for your pair(s)?

   Discuss benefits and disadvantages of regular re-estimation of cointegrated relationships by shifting data 1-2 weeks (remember to reserve some future data), and report not only on rolling β'_{Coint}, but also Engle-Granger Step 2, the history of value of test statistic for the coefficient in front of EC term.

   Would you implement something like Kalman filter/particle filter adaptive estimation [applied to cointegrated regression] in order to see the updated β'_{Coint} and μ_e? Reference: `www.thealgoengineer.com/2014/online_linear_regression_kalman_filter/`.

TS Project Workshop, Cointegration Lecture and Pairs Trading tutorial are your key resources.

Reading List: Cointegrated Pairs

- *Modeling Financial Time Series*, E. Zivot & J. Wang, 2002 – one recommended textbook, we distribute Chapter 12 on Cointegration with the relevant Project Workshop.
- Instead of a long econometrics textbook, read up *Explaining Cointegration Analysis: Parts I and II* by David Hendry and Katarina Juselius, 2000 and 2001. *Energy Journal*.
- Appendices of this work explain key econometric and OU process maths links, *Learning and Trusting Cointegration in Statistical Arbitrage* by Richard Diamond, *WILMOTT* `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2220092`.

Portfolio Construction using Black-Litterman Model and Factors

Summary

Construct a factor-bearing portfolio, compute at least two kinds of optimisation. Within each optimisation, utilise the Black-Litterman model to update allocations with absolute and relative views. Compute optimal allocations for three common levels of risk aversion (Trustee/Market/Kelly Investor). Implement systematic backtesting: which includes both, regressing results of your portfolio on factors and study of the factors themselves (wrt the market excess returns).

Kinds of optimisation: mean-variance, Max Sharpe Ratio, higher-order moments (min coskewness, max cokurtosis) – implement at least two. Min Tracking Error also possible but for that your portfolio choice will be measured against a benchmark index. Computation by ready formula or specialised for quadratic programming. Adding constraints improves robustness: most investors have margin constraints / limited ability to borrow / no short positions.

OPTIONALLY, Risk Contributions can also computed *ex ante* for any optimal allocation, whereas computing ERC Portfolio requires solving a system of risk budget equations (non-linear). ERC computation is not an optimisation, however can be 'converted' into one – sequential quadratic programming (SQP).

Portfolio Choice and Data

The choice of portfolio assets must reflect optimal diversification. The optimality depends on the criterion. For the max possible decorrelation among assets, it is straightforward to choose the least correlated assets. For exposure/tilts to factor(s) – you need to know factor betas *a priori*, and include assets with either high or low beta, depending on purpose.

A naive portfolio of S&P500 large caps is fully exposed to one factor, the market index itself, which is not sufficient. Specialised portfolio for an industry, emerging market, credit assets should have 5+ names, and > 3 uncorrelated assets, such as commodity, VIX, bonds, credit, real estate.

Factor portfolio is more of a long/short strategy, e.g., momentum factor means going long top 5 raising stocks and short top 5 falling. Factor portfolios imply rebalancing (time diversification) by design.

- mean-variance optimisation was specified by Harry Markowitz for simple returns (not log) which are *in excess* of the r_f. For risk-free rate, 3M US Treasury from pandas FRED dataset/ECB website rates for EUR/some small constant rate/zero rate – all are acceptable. Use 2-3 year sample, which means > 500 daily returns.
- Source for prices data is Yahoo!Finance (US equities and ETFs). Use code libraries to access that, Google Finance, Quandl, Bloomberg, Reuters and others. If benchmark index not available, equilibrium weights computed from the market cap (dollar value).
- In this variation of PC topic, it is necessary to introduce 2-3 factor time series and treat them as investable assets (5 Fama-French factors). If using Smart Beta ETFs present on their structure – you might find there is no actual long/short factors, just a long-only collection of assets with particularly high betas.

Step-by-Step Instructions

Part I: Factor Data and Study(Backtesting)

1. Implement Portfolio Choice based on your approach to optimal diversification. Usually the main task is to select a few assets that gives risk-adjusted returns the same as/outperforms a much larger, naturally diversified benchmark such as S&P500. See Q&A document distributed at the Workshop.

2. Experiment which factors you are going to introduce, collect their time series data or compute.

   - The classic Fama-French factors are HML (value factor) and SMB (small business). RMW (robust vs. weak profitability) and CMA (conservative vs aggressive capex) are the new factors and you can experiment with them.
   - Exposure to sector or style can also be considered a factor.
   - It very recommended that you introduce an interesting, custom factor such as Momentum, BAB (betting against beta) – likely you will need to compute time series of its returns, however that can be as simple as returns from a short portfolio of top five tech stocks.

3. The range of portfolios, for which factors are backtested, is better explained at source `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

4. Present P&L returns and Systematic Backtesting of your factors vs the Market (index of your choice), which includes performance, present plots of rolling beta and changing alpha. Ideally, you can present results for each factor beta independently and then, in combination. This work to be presented even before you engage in portfolio optimisation

Part II: Comparative Analysis of BL outputs

1. Plan your Black-Litterman application. Find a ready benchmark or construct the prior: equilibrium returns can come from a broad-enough market index. Implement computational version of BL formulae for the posterior returns.

2. Imposing too many views will make seeing impact of each individual view difficult.

3. Describe analytically and compute optimisation of **at least two kinds**. Optimisation is improved by using sensible constraints, eg, budget constraint, 'no short positions in bonds' but such inequality constraints $\forall w_i > 0$ trigger numerical computation of allocations. e.

4. You will end up with multiple sets of optimal allocations, even for a classic mean-variance optimisation (your one of two kinds). Please make your own selection on which results to focus your Analysis and Discussion – the most feasible and illustrative comparisons.

   - Optimal allocations (your) vs benchmark for active risk. Expected returns (naïve) vs implied equilibrium returns (alike to Table 6 in BL Guide by T. Idzorek.)
   - BL views are not affected by covariance matrix – therefore, to compute allocations shifted by views (through Black-Litterman model) with naive or robust covariance is your choice.

- Three levels of risk aversion – it is recommended that you explore at least for classical Min Var optimisation.

5. There is no rebalancing task for the project, particularly because posterior BL allocations expected to be durable.
6. Compare performance of your custom portfolio vs factors and market (rolling beta), independently and jointly. OPTIONALLY, compare performance of your portfolio to $1/N$ allocations / Diversification Ratio portfolio / Naive Risk Parity kind of portfolio and perform the systematic backtesting of that portfolio *wrt* to factors.

Reading List

- CQF Lecture on Fundamentals of Optimization and Application to Portfolio Selection
- A Step-by-step Guide to The Black-Litterman Model by Thomas Idzorek, 2002 tells the basics of what you need to implement.
- The Black-Litterman Approach: Original Model and Extensions Attilio Meucci, 2010. http://ssrn.com/abstract=1117574
- On LW nonlinear shrinkage / Marcenko-Pastur denoising, either method to make a covariance matrix robust, resources and certain code provided with the relevant Workshop and Tutorial.

Optimal Hedging with Advanced Greeks

Summary

In this topic, you first consider the simple volatility arbitrage under condition of future realised volatility to be above the implied $V_a > V_i$. The workings can be found in Understanding Volatility lecture and solutions. You have to implement in code the delta replication (long option, short stock) using high/medium/low volatility values of your choice. Use European call option Black-Scholes formulae. Provide visibility into how Gamma affects the ongoing P&L. It is not necessary to consider a portfolio of options or several assets – the task is simpler than that.

Improvement in delta-hedging can be achieved by adjusting the naive Black-Scholes Delta. We recommend to follow Minimum Variance Delta method and compute the adjustment that takes into account the expected changes in σ_{imp} as a result of changes in asset S_t. For that you need to numerically compute Vega and $\frac{\partial E(\sigma_{imp})}{\partial S}$. Under the hood, MV Delta computation relies on quadratic fit to implied volatilities, simple but appropriate application. MV Delta does not need mixing with the local volatility in this project, though it easily allows that.

Greeks estimated from advanced models are superior to Δ_{BS} which relies on the implied vol of the day. In derivatives trading with path-dependent option products, the local volatility (LV) often takes over stochastic volatility (SV) despite the latter's explanatory powers. The associated benefits of LV and SV and more recently Rough Volatility (RV) modeling stem from them overcoming Black-Scholes limitations (as in the lectures) and conformity with volatility smile present in markets.

The first purpose here is to numerically confirm the relationship between Implied Volatility and Local Volatility, for a simplified vol surface. That relationship is theoretically well-known but you need to re-discover it for yourself. For the local volatility stripper use a ready library/code initially. To re-implement from the first principles industry-level numerical techniques of estimating LV from smoothed IV surface – is a very advanced task and not a requirement. However, you can use Dupire local volatility explicitly in order to confirm the correctness of $\sigma_{LV}(K, T)$ at several points of the surface.

Instrument Choice and Data. In this project, you will be generating equity prices by Monte-Carlo. In the absence of Bloomberg/professional data for the option prices and volatility surfaces, please sensibly generate the data yourself. There is only one rule: exercise care when assuming volatility numbers to avoid inconsistencies the term structure – see σ_t bootstrapping in Understanding Volatility. It does not matter whether ATM asset volatility is 15% or 45%. You can sensibly confirm the project outcomes about P&L and relate to real-life option markets.

The stripping of local volatility needs numerically smooth implied volatility surface that is, having quadratic or cubic fittings are preferable over a matrix of vanilla options of different maturities and strikes. However, to confirm $\sigma_{LV}(K, T)$ you only need to consider $\pm 3 - 4$ strikes In- and Out-of-Money, in addition to ATM.

Step-by-Step Instructions

Part I: Simple Volatility Arbitrage but improved Asset Evolution

1. Consider improvements you can add to the standard Monte Carlo for GBM asset evolution. That can include Euler-Maruyana/Milstein schemes. Monte Carlo improvement is not trivial to the task.
   - consider variance reduction techniques, such as antithetic variates;
   - best practice is low-discrepancy sequences such as Sobol with the Brownian bridge.
2. For volatility arbitrage under condition of known future realised volatility $V_a > V_i$, analytically and with Monte-Carlo confirm the items below. In the report present both, complete mathematical workings to fold $P\&L_t$ and simulations of $P\&L_t$.
   - Confirm that Actual volatility hedging leads to the known total P&L.
   - Confirm and and demonstrate that Implied volatility hedging leads to **uncertain** path-dependent total P&L.
3. Think of additional analysis you can provide, consider how P&L decomposes in terms of Greeks. What is the impact of time-dependent Gamma Γ_t? What about $r^2 - \sigma_{imp}\delta t$?

Part II: Advanced Greeks

1. Recompute the scenarios/cases from simple volatility arbitrage (Part I) now using Minimum Variance Delta.
   - numerically compute the adjustment for expected changes in σ_{imp} as a result of changes in asset price S_t, which is $\frac{\partial E(\sigma_{imp})}{\partial S}$.
   - do quadratic fitting on ATM term structure of options – implied volatility. Fit coefficients (parameters) a, b, c can be constant for a study project, however the original Hull research uses rolling regression. Decide on frequency of re-fitting, eg 5-10 working days (even 22) and the project would not need much data. Use the history of term structure strips of ATM options, or reasonably generate pseudo historical data!. To reiterate, you don't need to work with full vol surface from each day.
2. Our model validation will be limited to the analysis of change of a, b, c and $\sigma_{MV} - \sigma_{BS}$ over time, produce plots, measure a squared error if the parameter seems stationary.

Part III: Local Volatility addition

Approach A Prepare data for the local volatility stripper (ready library or code) of your choice. Take real-life or reasonably simulated implied volatility surface and compute one-off local volatility surface.

- Demonstrate good understanding of the maths and principles of the local vol stripper and present on it mathematically in the report (without re-coding). You can double-check (validate) the stripper with Dupire result.
- Confirm the relationship between Implied Volatility and Local Volatility for a simplified vol surface. The relationship should conform to $\Sigma_{imp}(S,K) \approx \sigma(S) + \frac{\beta}{2}(K-S)$.

Approach B Take (or make up) a simplified implied vol surface, basically a matrix of vanilla options of different maturities and strikes, and use Dupire local volatility to compute the correctness of $\sigma_{LV}(K,T)$.

- Confirm the relationship between Implied Volatility and Local Volatility for a simplified vol surface. The relationship should conform to $\Sigma_{imp}(S,K) \approx \sigma(S) + \frac{\beta}{2}(K-S)$

1. OPTIONAL if your time permits, test the performance of delta computed using $\sigma_{LV}(K,T)$ in the simple volatility arbitrage setup $V_a > V_i$. The task is simple but you need the stripped local volatility at each time step (or say each 5 or 22 days). It seems data-intensive, however you only need to test for ATM local vol and decide whether to utilise the short-term or longer-term vol.

Module 3 Lectures, Project Workshop and Tutorials are your key resources. If experiencing lack of understanding please print and review the key papers in Additional Material.