Certificate in Quantitative Finance
Final Project Brief

JAN 2022 COHORT

This document outlines each available topic together with submission requirements for the project report and code. By-step instructions offer a structure not a limit to what you can implement. Project Workshops I and II will provide focus to the implementation of each topic.

CQF Final Project is numerical techniques and backtest or sensitivity-test of model output (prices, portfolio allocations, pairs trading) as appropriate. Some numerical techniques call for being implemented in code from first principles. There might be numerical methods either too involved or auxiliary which you would not need to implement if good ready functionality is available. Re-use and adoption of code permitted. Marks earned will strongly depend on coding of numerical techniques and presentation of how you explored and tested a quantitative model.

A capstone project require a degree of own study and ability to work with documentation on packages that implement numerical methods in your coding environment eg, Python, R, Matlab, C#, C++, Java, Scala, VBA+NAG libraries.

Exclusively for current CQF delegates. No distribution.

Dr Richard Diamond	PC, TS
Dr Riaz Ahmad	CR, Python/C++
Kannan Singaravelu	DL, Python
CQF Team	All submission questions

This brief prepared and overall final project supervision is by Dr Richard Diamond.

1 Topics

To complete the project, you must implement **one topic** from this list – according to the tasks of up to date Brief. 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. **Portfolio Construction using Black-Litterman Model and Factors - PC**
2. **Deep Learning for Financial Time Series - DL**
3. **Long/Short Trading Strategy Design & Backtest - TS**
4. **Credit Spread for a Basket Product - CR**

1.1 Project Report and Submission Requirements

- Submit working code together with a well-written report and originality declaration.
- There is no set page length. Report must have an analytical quality and discussion of results/robustness/sensitivity/backtesting as appropriate to the topic.
  Use charts, test cases and comparison to empirical research papers where available.
- Report must contain sufficient mathematical model, numerical methods with attention to their convergence/accuracy/computational properties.
  Please feature the numerical techniques you coded – make a table.
- Mathematical sections can be prepared using LaTeX or Equation Editor (Word). Printing out Python notebook code and numbers on multiple pages, without your own analysis text, explained plots, relevant maths is not an acceptable format.

Must be submitted by 23:59 BST on submission date.

Work done must match the Brief. There is no extension to the Final Project.

Projects without declaration or working code are incomplete and will be returned.

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

1.2 Advanced Electives

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 longer immediate match between Project Topics and Electives.
- Several viable combinations for each Project Topic are possible.
- One effective learning strategy is to select one 'topical elective' and one 'coding elective'.
- At later stage, all Advanced Electives will become unlocked – available later for your advancement.
- Please review the descriptions, table below and choose two Electives for your interest and needs. At the stage of Project Brief, the tutors not have other advice than descriptions presented.

The detailed and up to date description of electives is availalbe from `https://www.cqf.com/about-cqf/program-structure/cqf-qualification/advanced-electives`

Counterparty Credit Risk – choose for CR topic	CDS, survival probabilities and hazard rates reviewed. Three key numerical methods for quant finance pricing (Monte-Carlo, Binomial Trees, Finite Difference). Monte Carlo for simple LMM. Review of Module Five on Credit with a touch on the copula method. **Outcome:** covers CVA Computation clearly and reviews of credit spread pricing techniques.
Risk Budgeting – choose for PC topic	Reviews the nuance of Modern Portfolio Theory, ties in VaR and Risk Decomposition with through derivations and expectation algebra. Gives simple examples of figures you need to compute and then combine with portfolio optimisation. Risk-budgeting portfolio from Video Part 10.
Adv Risk Management – useful in general	The base you need to be a risk manager (read Coleman guide) and tools for Basel regulation: (a) weakness of risk-weighted assets, (b) extreme value theory for ES and capital requirement and (c) adjoint automatic differentiation to compute formal sensitivities. Other numericals covered are the same as Counterparty Risk. **Outcome:** this elective is best taken for your own advancement.
Behavioural Finance – optional, PC and ML	Heuristics, biases and framing. An excursion to game theory and probability to formalise the psychology of choice.

Table 1: Topical Electives – Primarily Supporting the Final Project

The main source of instruction for the projects remains this Brief, Project Workshops, and Additional Material – a small, curated collection of articles offered for each topics and distributed with Workshops. Please also see Reading Lists.

1.3 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 – doesn't mean your project to be based on that ready code.
- Python with pandas, matplotlib, sklearn, and tensorflow forms a considerable challenge to Matlab, even for visualisation. 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, <u>reflect on the issue in the project report</u>.
- 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.

This page intentionally left blank.

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 <u>at least two</u>. 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.

Deep Learning for Financial Time Series

Summary

A recurrent neural network can be a successful alternative to time series regression if one believes the data carries autoregressive structure. For this topic you will run Long Short-Term Memory classification (up/down moves) with features more advanced than ML Assignment (Exam 3). Volatility can be a feature itself, together with an interesting addition of drift-independent volatility. One specific model of recurrent NN is Long Short Term Memory, it can come out as one of best possible predicting models from features, such as:

- financial ratios/adv technical indicators/volatility estimators.
- OPTIONALLY if you can access data, you can enhance prediction with (a) credit spreads – traded CDS or indices and (b) news indicators – FactSet, RavenPack offer API for professional subscription or trial.

Dealing with the arbitrary length of sequence is the major characteristic of LTSM – that means you can attempt to use frequency longer than 1D. Certain time series, such as interest rates or economic indicators are more characterised by long memory and stationarity, and therefore modelled with power law autocorrection/Hurst exponent fractional/Markov processes. If you attempt the prediction of 5D or 10D return for equity or 1W, 1M for FF factor the challenge is twofold. First is increase data requirement, nearing 7-10 years to begin with. Second is isolation of autocorrelation in positive 5D/10D return in equity time series.

Before using RNNs, conduct exploratory data analysis (EDA) in order to create features map (not tickers map highlighted in tutorial).[1] You can proceed straight to autoencoding for NNs or utilise self-organising maps (SOM)/decision tree regressor (DTR). Multi scatterplots presenting relationships among features is always a good idea, however if initially you have 50-70 features (including repeated kind, different lookback) – you need to reduce dimensionality in features. SOM analysis can be on non-lagged features, daily data and lookback computational periods of 5D/10D/21D/50D for features as appropriate – immediately generates 4 columns of the same feature – and dense map areas should help you to answer such questions as to choose EMA 5D or 10D, Volatility 50D or ATR 10D. Alternatively, top-level splits of DTR with large numbers in leaf should reveal which features to prioritise.

Part I: Features Engineering

Please revisit ML Lab II (ANNs) for basic discussion on feature scaling. Be careful about sklearn feature selection by F-test.

1. Past moving averages of the price, simple or exponentially weighted (decaying in time), so SMA, EMA. Technical indicators, such as RSI, Stochastic K, MACD, CCI, ATR, Acc/Dist). Interestingly, Bollinger Bands stand out as a good predictor. Remember to vary lookback period

[1] EDA helps dimensionality reduction via better understanding of relationships between features, uncovers underlying structure, and invites detection/explanation of the outliers. EDA should save you from 'brute force' GridSearchCV runs calling a NN/RNN Classifier again and again.

5D/10D/21D/50D, even 200D for features as appropriate. Non-overlapping periods means you need data over long periods.

Volume information and Volume Weighted Average Price appear to be immediate-term signals, while we aim for prediction.

2. Use of features across assets are permitted but be tactical about design: eg, features from commodity price impacting an agricultural stock (but not oil futures price on an integrated oil major), features from cointegrated equity pair. Explore Distance Metrics among features (KNN) and potential K-means clustering as yet another alternative to SOM.
3. Balance the needs of longer-term prediction vs. short-term heteroskedastic volatility. Yang & Zhang (2000) provide an excellent indicator: drift-independent volatility which takes into account an overnight jump – but might not be as useful for 5D/long-term prediction as can't be re-scaled to non-daily jumps. Smoothed volatility estimate (EWMA/EGARCH) can be scaled $\sqrt{t}$ but it's not intended as a medium-term prediction indicator, and at the same time risks being dominated by the long-term average variance $\bar{\sigma}^2$
4. OPTIONAL Interestingly credit spreads (CDS) can be a good predictor for price direction. Think out of the box what other securities have 'credit spread' affecting their price.
5. OPTIONAL Historical data for financial ratios is good if you can obtain the data via your own professional subscription. Other than that, history of dividends, purchases/disposals by key stakeholders (director dealings) or by large funds, or Fama-French factor data is better available.

Part II: Pipeline Formation (considerations)

- Your implementation is likely be folded into some kind of ML Pipeline, to allow you re-use of code (eg, on train/test data) and aggregating the tasks. Ensemble Methods present an example of such pipleline: Bagging Classier is an umbrella name for the process of trying several parametrisations of the specific classifer (eg Logistic Regression). AdaBoost over Decision Tree Classifier is another case. However, please do not use these for DL Topic.
- Empirical work might find that RNNs/Reinforcement Learning might work better WITHOUT past returns! Alternatively, if you are predicting 5D/10D move there will be a significant autocorrelation effect – your prediction will work regardless of being a good model or not.
- Please limit your exploration to 2-3 assets and focus on features, their SOM (if possible), and LTSM Classifier to make the direction prediction. If you are interested in the approach to choose a few from a large set of assets – can adopt a kind of diversified portfolio selection (see Portfolio Topic Q&A).
- You are free to make study design choices to make the task achievable. Substitutions:
  - present relationship between features with simple scatterplots (vs SOMs) or K-means clustering;
  - use MLP classifer if recurrent neural nets or LTSM is particular challenge;
  - re-define task and predict Momentum sign (vs return sign) or direction of volatility.

Under this topic you do not re-code decision trees or optimisation to compute NNs weights and biases.

Pairs Trading Strategy Design & Backtest

Estimation of cointegrated relationship between prices allows to arbitrage a mean-reverting spread. Put trade design and backtesting in the centre of the project, think about your signal generation and backtesting of the P&L. You will have a hands-on experience with regression but will not run the regression on returns. The numerical techniques are regression computation in matrix form, Engle-Granger procedure, and statistical tests. You are encouraged to venture into A) multivariate cointegration (Johansen Procedure) and B) robustness checking of cointegration weights, ie, by adaptive estimation of your regression parameters with statistical filters.

Contegrating weights that you use to enter the position form the long/short allocation that produces a mean-reverting spread. Signal generation and suitability of that spread for trading depend on its fitting to OU process recipe. For optimisation, comparative backtesting, rolling ratios and other industry-level backtesting analytics use the ready code libraries. However, project that solely runs pre-programmed statistical tests and procedures on data is insufficient. It is not recommended using VBA for this topic due to lack of facilities.

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 you can use own understanding of multivariate cointegration analysis and R/Python VECM packages in order select the best candidates for pairs trading (or even basket trading). Do not use all five Zivot's 'deterministic trends' in coint residual – in practice one only need a constant inside the residual e_{t-1}.

Part II: Backtesting

It is your choice as a quant to decide which elements you need to argue successfully that your trading strategy (a) will not fall apart and (b) provides 'uncorrelated return'.

4. Perform Systematic Backtesting of your trading strategy (returns from pairs/basket trading) platform to produce drawdown plots, rolling Sharpe Ratio and rolling beta (with regard to market and at least one factor/industry of your choice).
5. Industry backtesting relies on rolling betas, while scientific research will test for breakouts using LR test. One hand, cointegrated relationship supposed to persist and β'_{Coint} should stay the same. Keep delivering stationary spread over say, 3-6 months, without the need to be updated. However, Kalman filter/particle filter adaptive estimation of coint regression will give updated β'_{Coint} and μ_e. `http://www.thealgoengineer.com/2014/online_linear_regression_kalman_filter/`.

   However, you can simply re-estimate 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.
6. Use other practices of industry and Machine Learning-inspired backtesting, such as splitting data into train/test subsets, preprocessing, and crossvalidaiton (as appropriate and if feasible).

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

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 $\pm$ 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`

Relevant Readings 2022-Jan

The list puts together some initial – you will find more topical resources inside Additional Material.zip, provided for each Topic within the relevant Project Workshop I or II, or within the anchor core lecture, such as on Neural Nets. Please do not email tutors for a copy.

Reading List: Credit Portfolio

- 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.

Reading List: Portfolio Construction

- 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.

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`.